Copyright Apdos

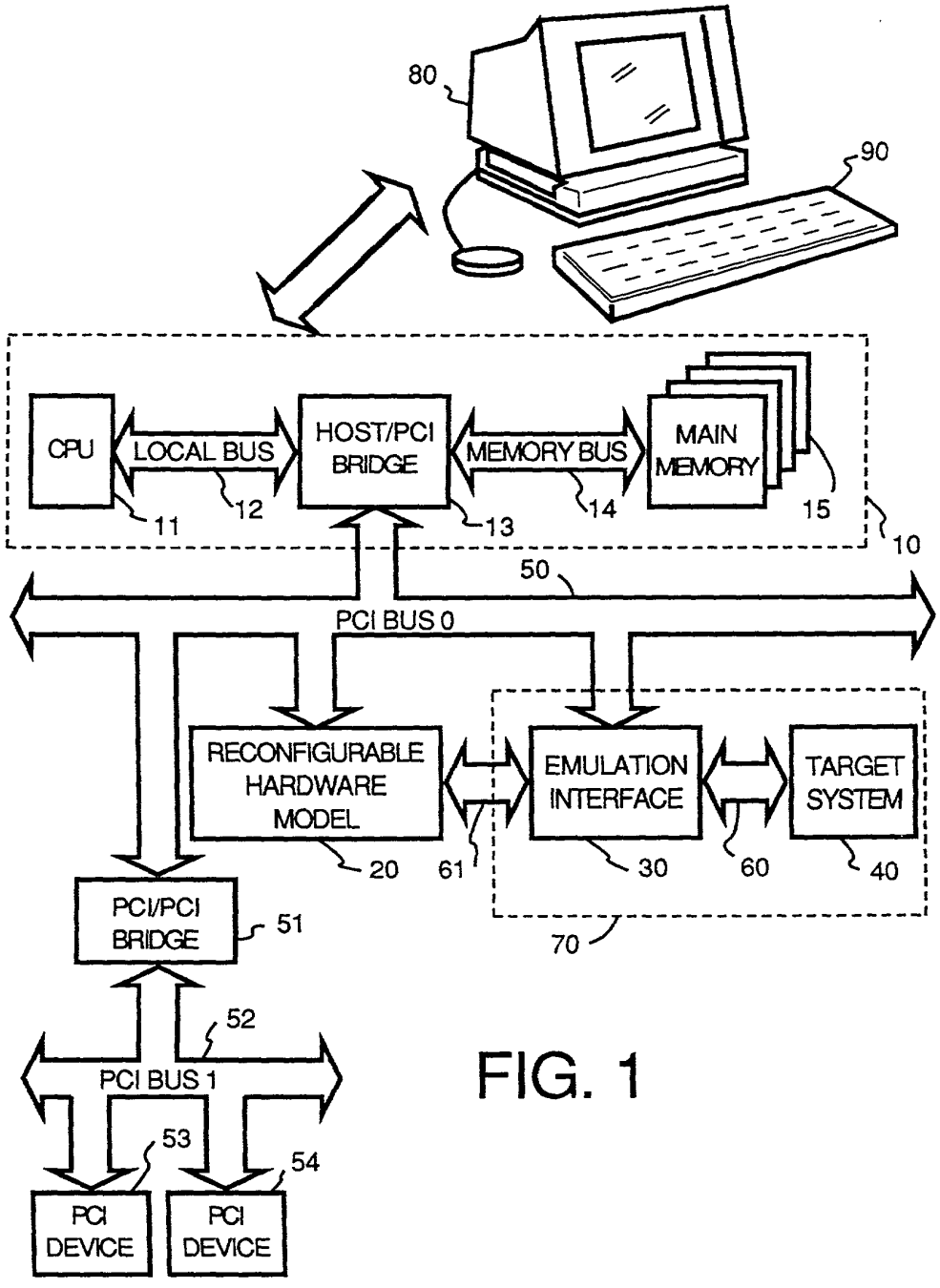


FIG. 1

EXHIBIT 1029

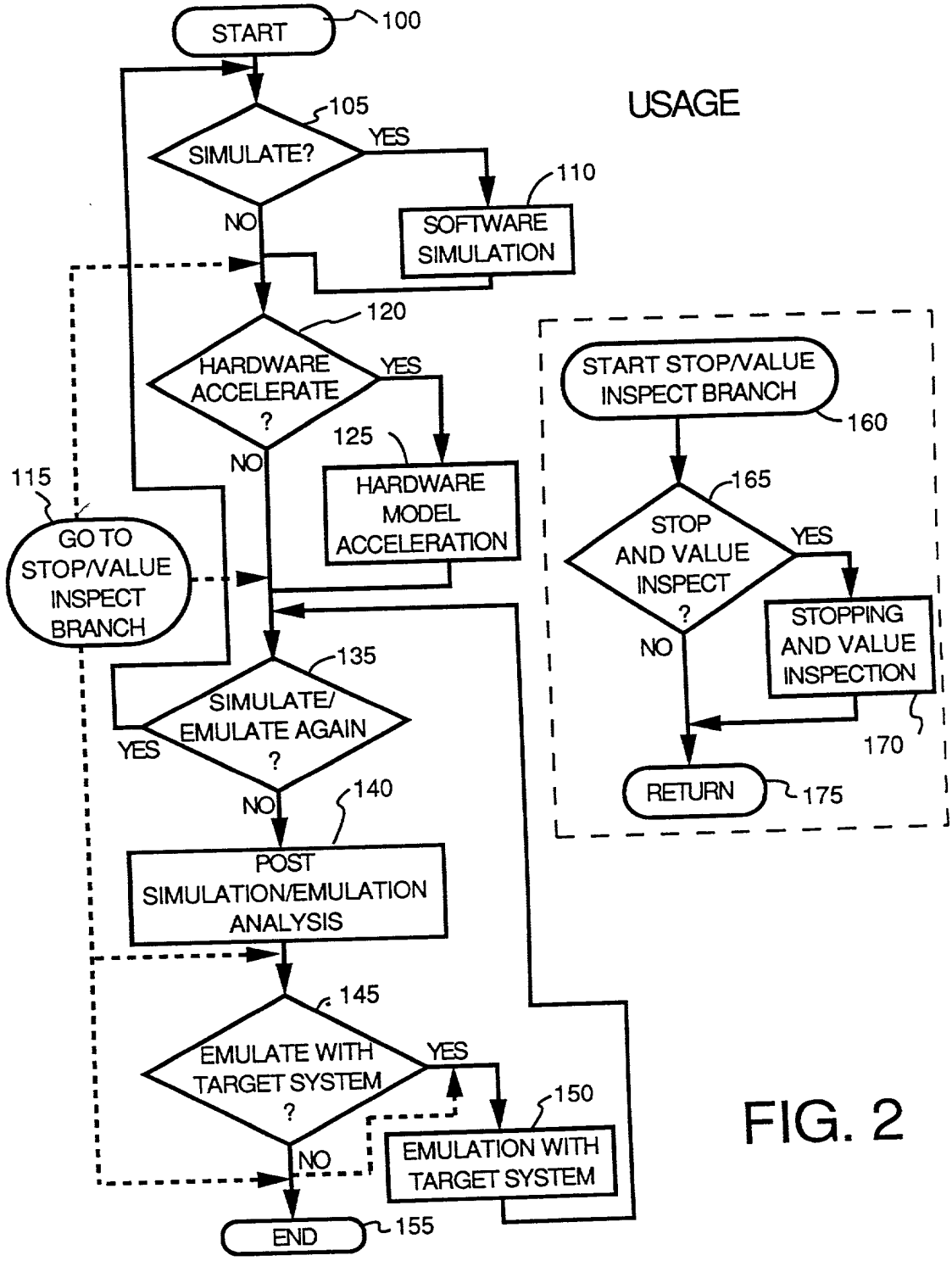


FIG. 2

TOOTOT"hZTOO660

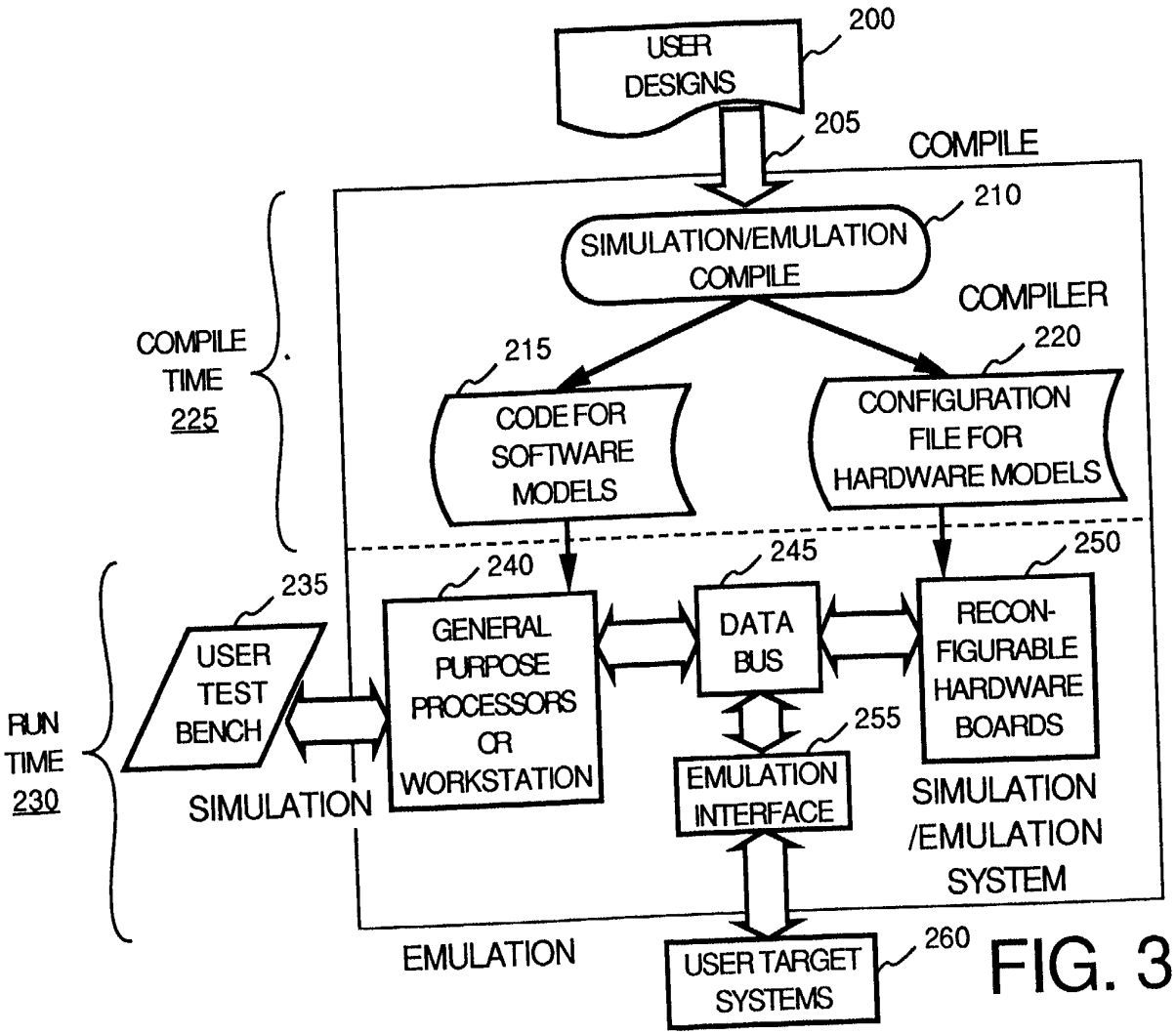


FIG. 3

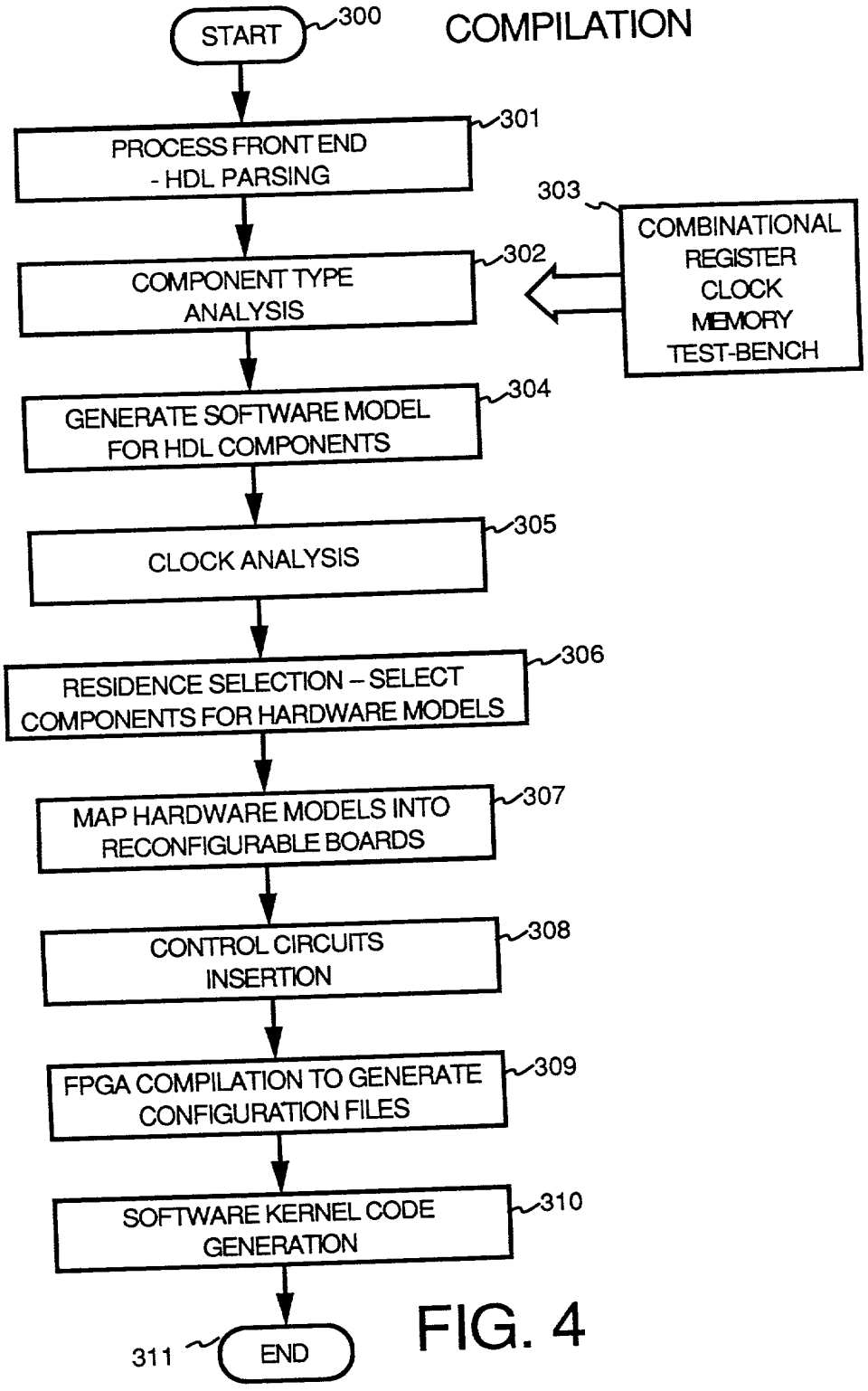

COMPILATION
START
300
PROCESS FRONT END
- HDL PARSING
301
303
COMBINATIONAL
REGISTER
CLOCK
MEMORY
TEST-BENCH
COMPONENT TYPE
ANALYSIS
302
GENERATE SOFTWARE MODEL
FOR HDL COMPONENTS
304
CLOCK ANALYSIS
305
RESIDENCE SELECTION – SELECT
COMPONENTS FOR HARDWARE MODELS
306
MAP HARDWARE MODELS INTO
RECONFIGURABLE BOARDS
307
CONTROL CIRCUITS
INSERTION
308
FPGA COMPILATION TO GENERATE
CONFIGURATION FILES
309
SOFTWARE KERNEL CODE
GENERATION
310
311
END

FIG. 4

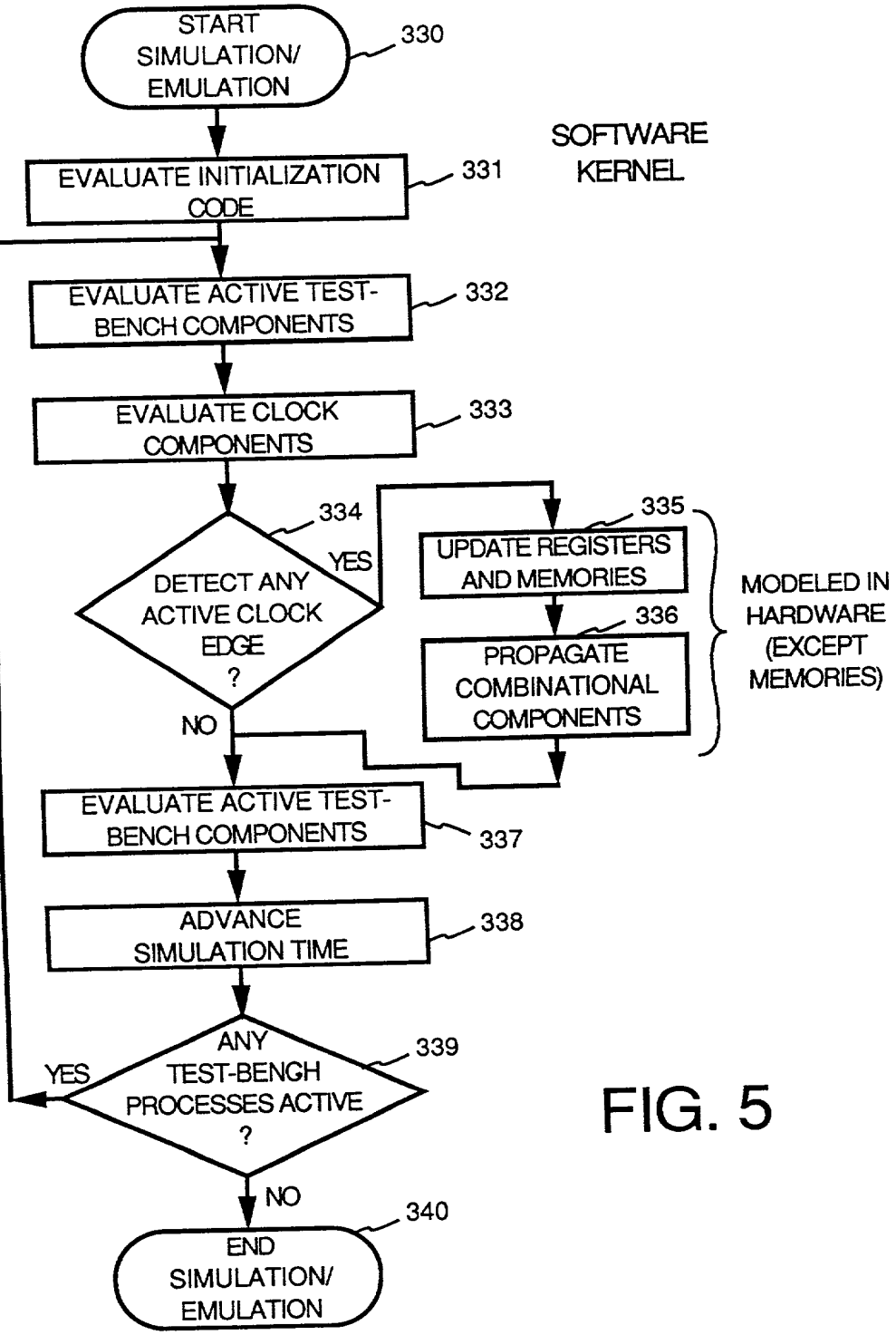

START SIMULATION/ EMULATION
330
SOFTWARE KERNEL
EVALUATE INITIALIZATION CODE
331
EVALUATE ACTIVE TEST-BENCH COMPONENTS
332
EVALUATE CLOCK COMPONENTS
333
334
DETECT ANY ACTIVE CLOCK EDGE ?
YES
NO
335
UPDATE REGISTERS AND MEMORIES
336
PROPAGATE COMBINATIONAL COMPONENTS
MODELED IN HARDWARE (EXCEPT MEMORIES)
EVALUATE ACTIVE TEST-BENCH COMPONENTS
337
ADVANCE SIMULATION TIME
338
ANY TEST-BENCH PROCESSES ACTIVE ?
339
YES
NO
340
END SIMULATION/ EMULATION

FIG. 5

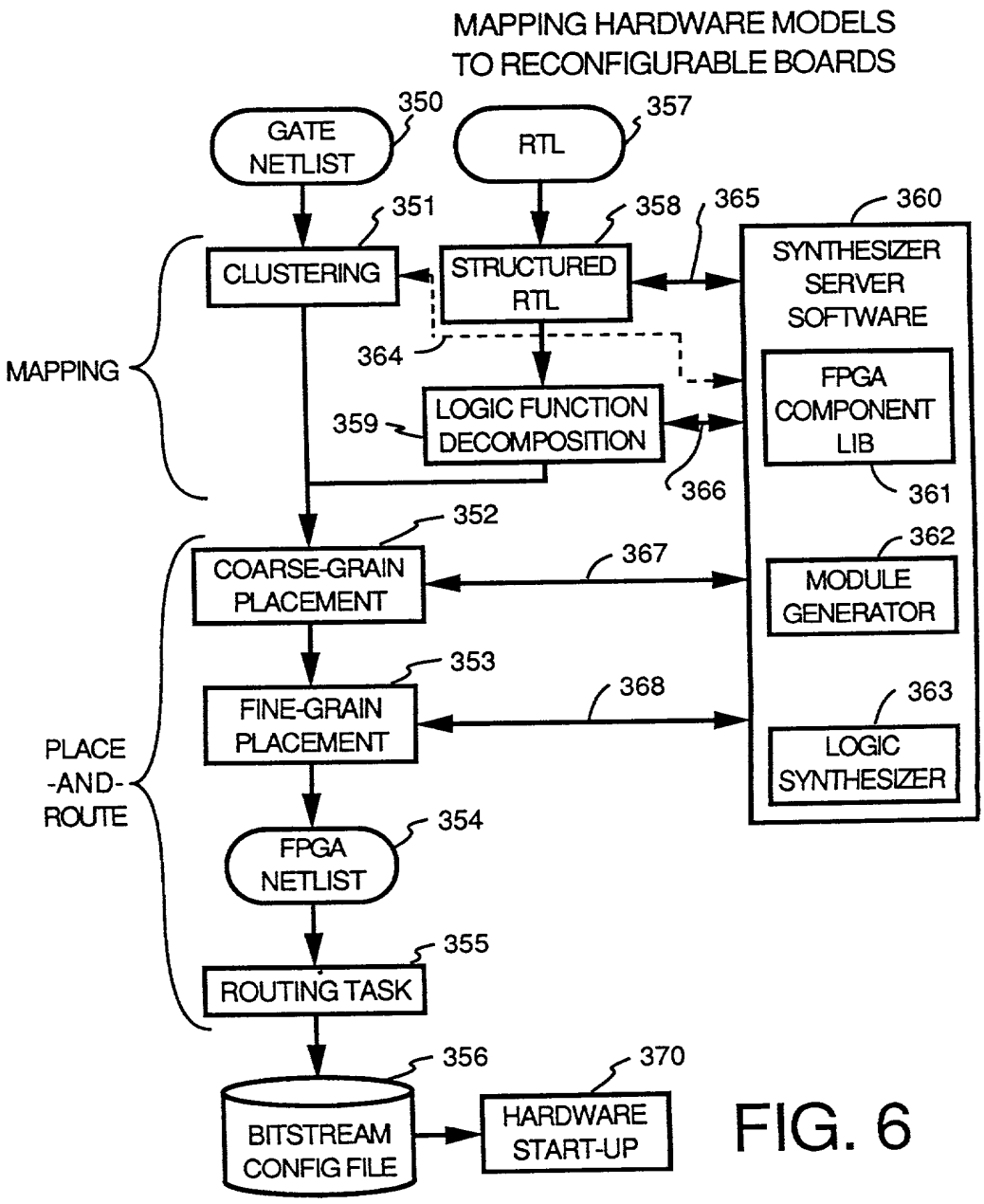

MAPPING HARDWARE MODELS
TO RECONFIGURABLE BOARDS
GATE NETLIST
350
RTL
357
CLUSTERING
351
STRUCTURED RTL
358
365
360
SYNTHESIZER SERVER SOFTWARE
MAPPING
364
FPGA COMPONENT LIB
LOGIC FUNCTION DECOMPOSITION
359
366
361
352
COARSE-GRAIN PLACEMENT
367
362
MODULE GENERATOR
353
FINE-GRAIN PLACEMENT
368
363
LOGIC SYNTHESIZER
PLACE -AND- ROUTE
FPGA NETLIST
354
ROUTING TASK
355
356
370
BITSTREAM CONFIG FILE
HARDWARE START-UP

FIG. 6

EP 0 700 160 A2

	F11	F12	F13	F14	F21	F22	F23	F24	F31	F32	F33	F34	F41	F42	F43	F44
F11	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0
F12	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0
F13	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0
F14	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1
F21	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0
F22	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0
F23	0	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0
F24	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1
F31	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0
F32	0	1	0	0	0	1	0	0	1	1	1	1	0	1	0	0
F33	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	0
F34	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
F41	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1
F42	0	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1
F43	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	1
F44	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1

FIG. 7

TOOTOT" h2T00660

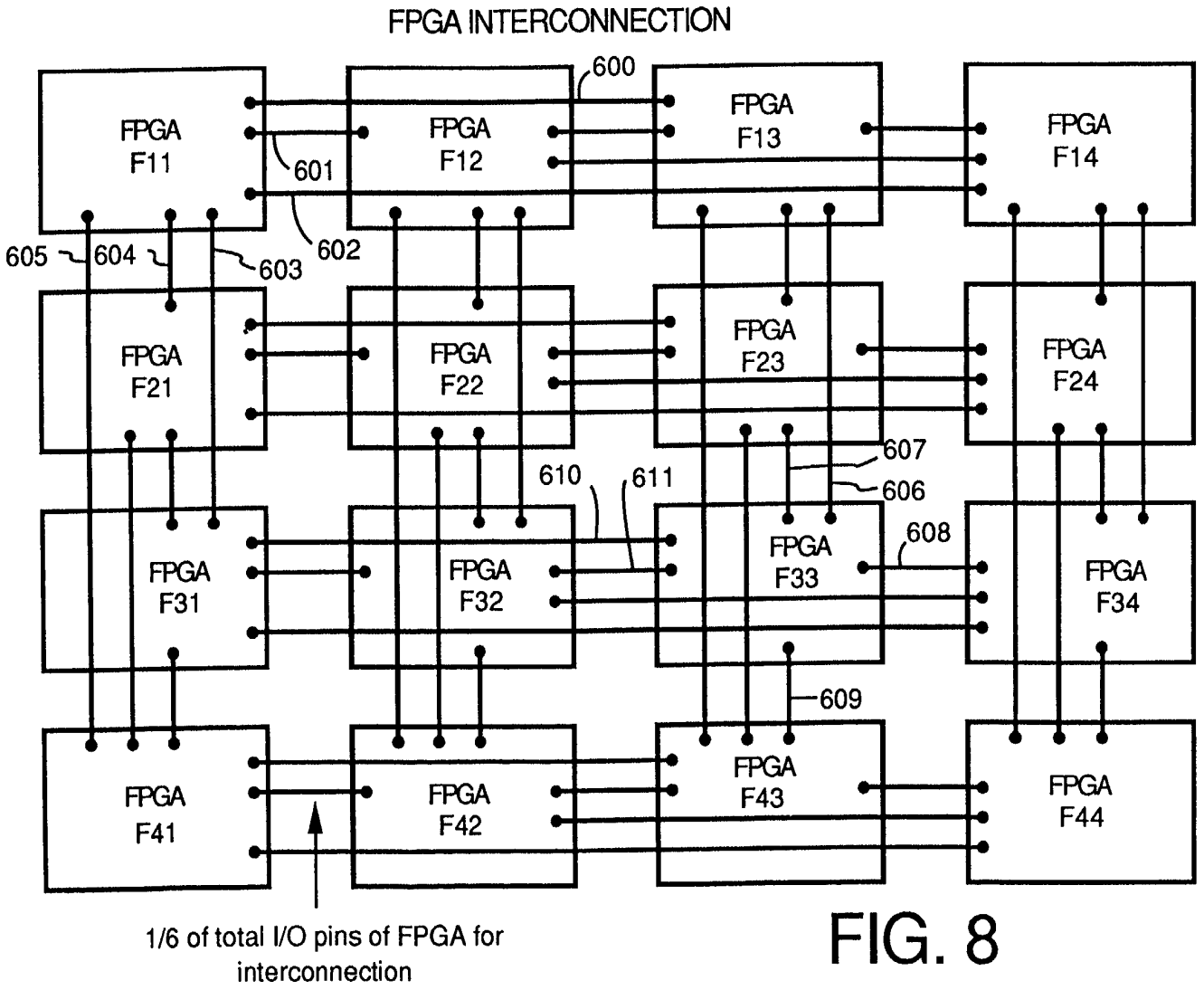


FIG. 8

T00T0T"h2T00660

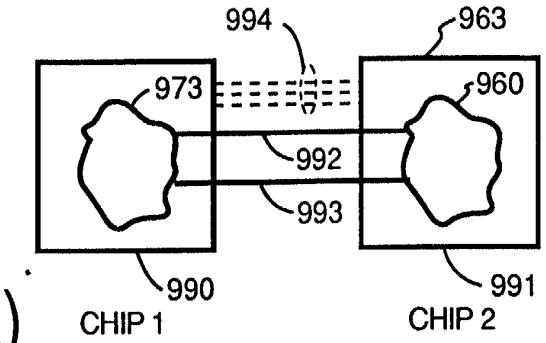


FIG. 9(a)

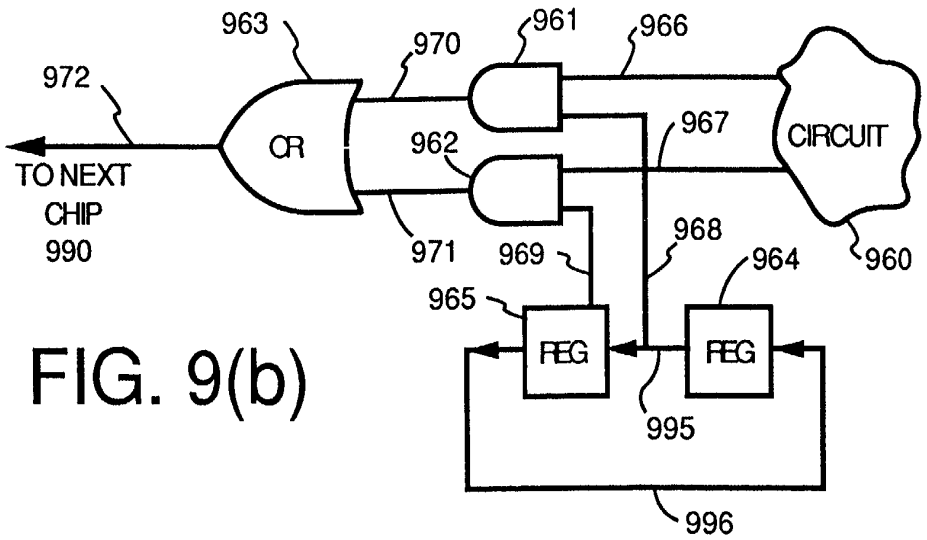


FIG. 9(b)

TOOTOT"h2T00660

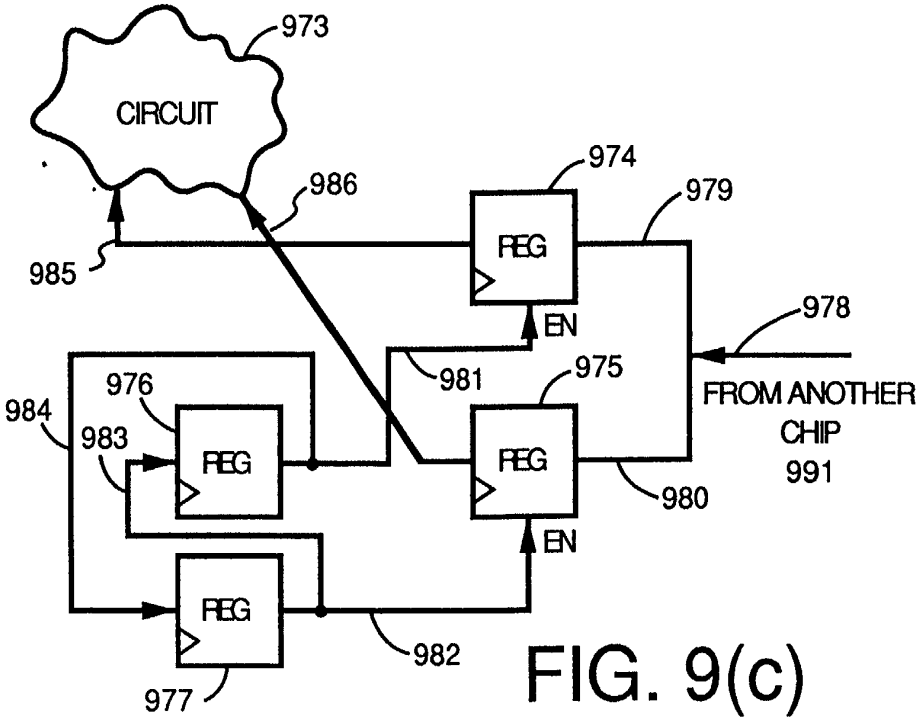


FIG. 9(c)

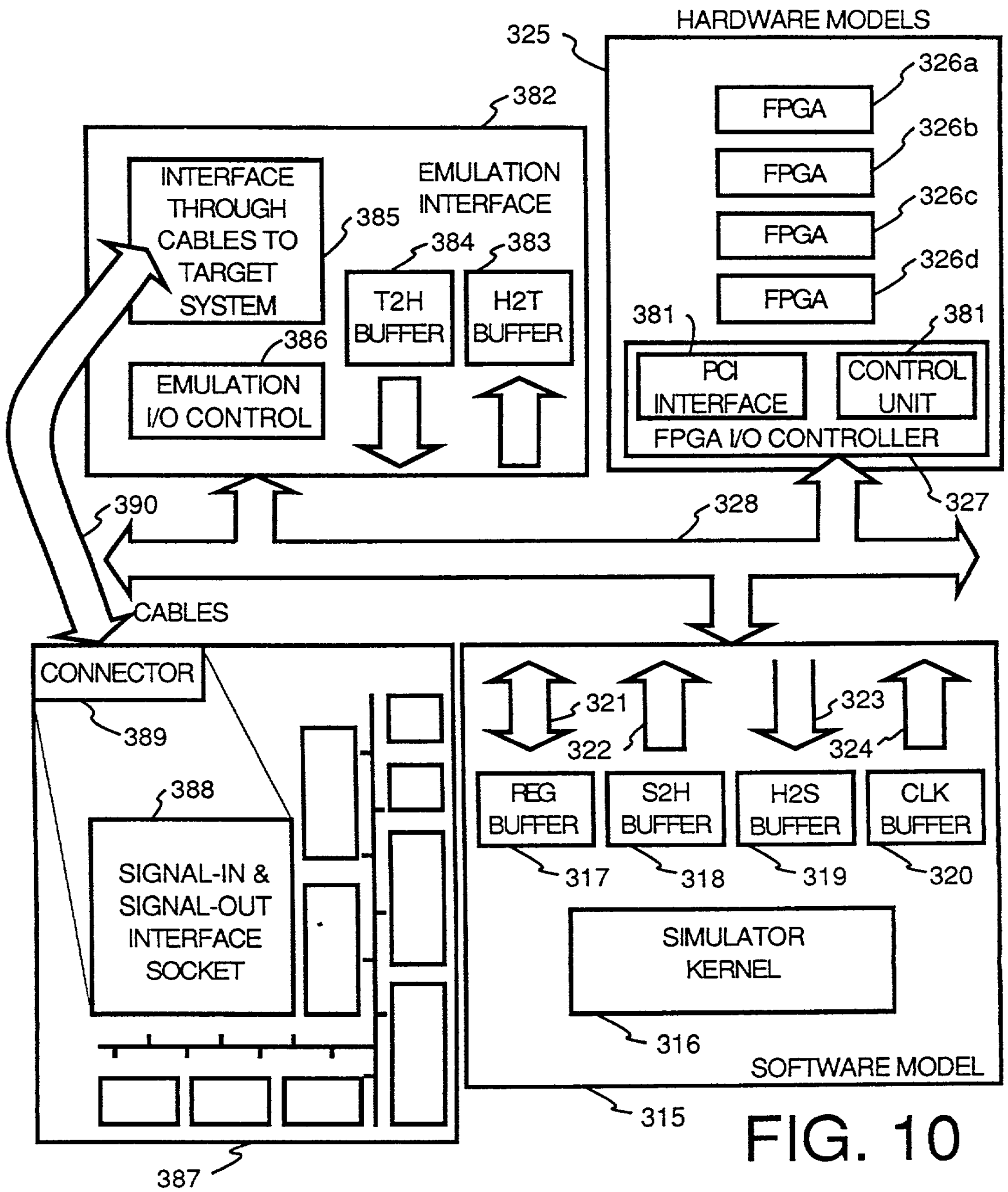

HARDWARE MODELS
325
326a
FPGA
326b
FPGA
326c
FPGA
326d
FPGA
381
381
PCI INTERFACE
CONTROL UNIT
FPGA I/O CONTROLLER
327
382
INTERFACE THROUGH CABLES TO TARGET SYSTEM
EMULATION INTERFACE
385
384
383
T2H BUFFER
H2T BUFFER
386
EMULATION I/O CONTROL
390
328
CABLES
CONNECTOR
389
388
SIGNAL-IN & SIGNAL-OUT INTERFACE SOCKET
387
321
322
323
324
REG BUFFER
S2H BUFFER
H2S BUFFER
CLK BUFFER
317
318
319
320
SIMULATOR KERNEL
316
SOFTWARE MODEL
315

FIG. 10

TOOTOT "h2T00660

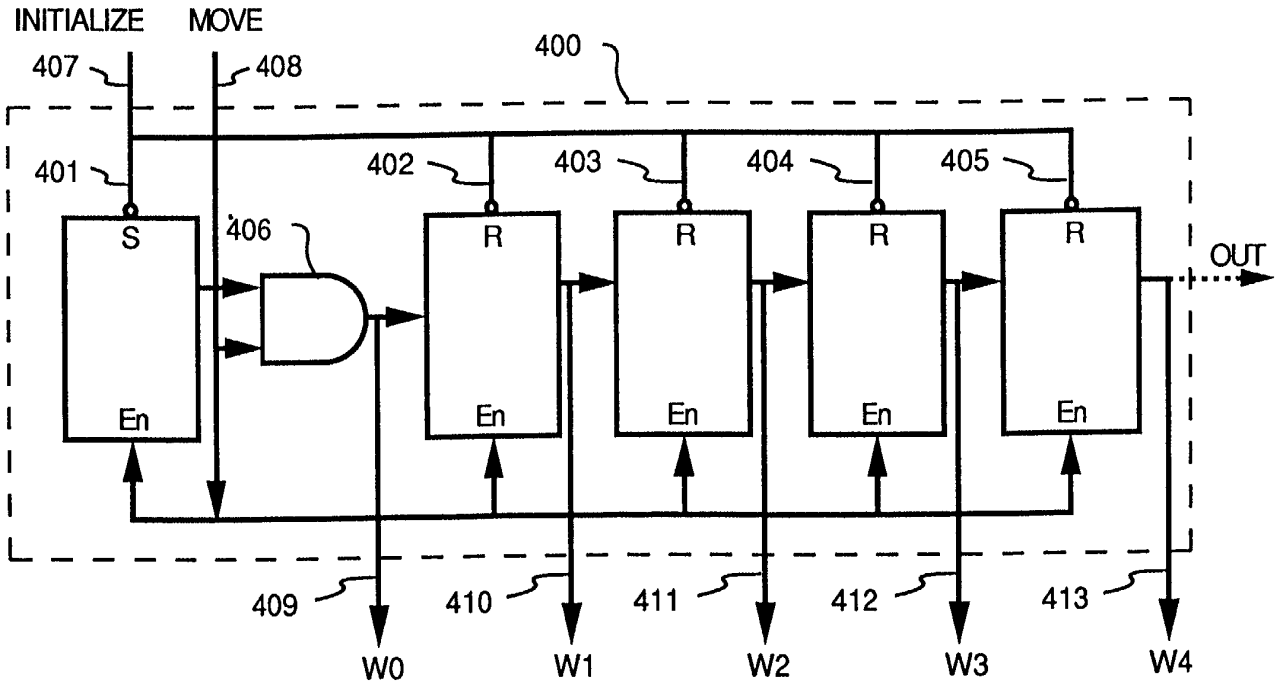


FIG. 11

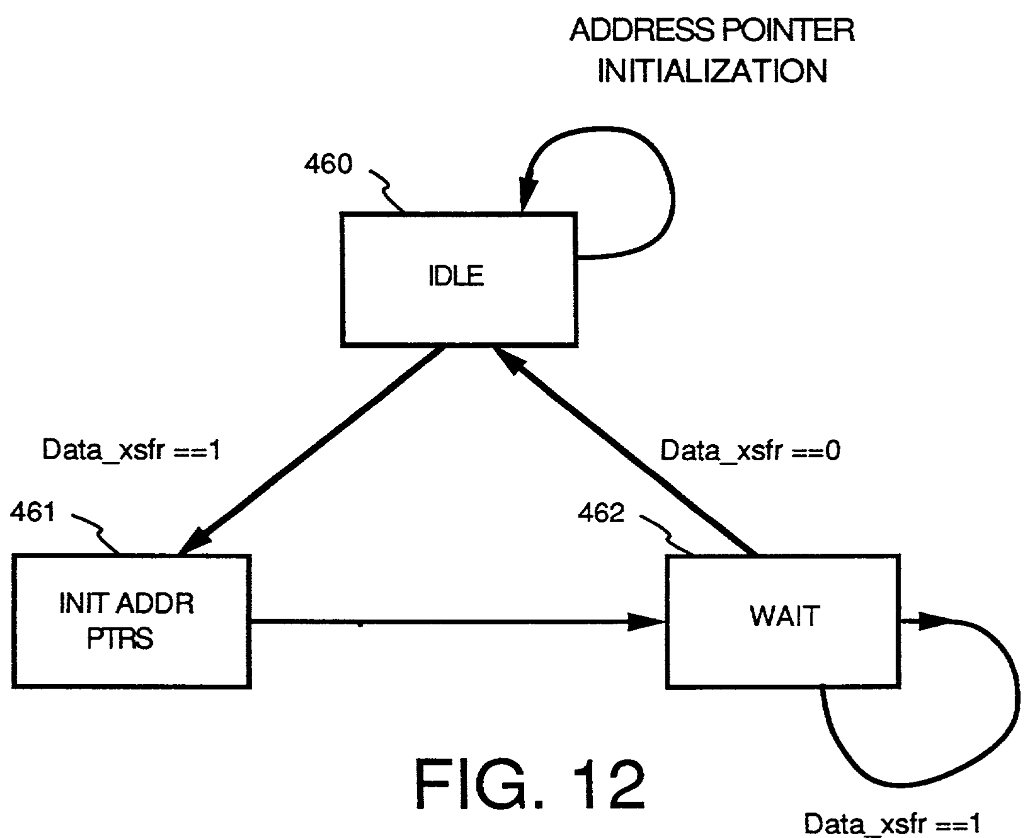

ADDRESS POINTER
INITIALIZATION
460
IDLE
Data_xsfr ==1
Data_xsfr ==0
461
462
INIT ADDR
PTRS
WAIT
Data_xsfr ==1

FIG. 12

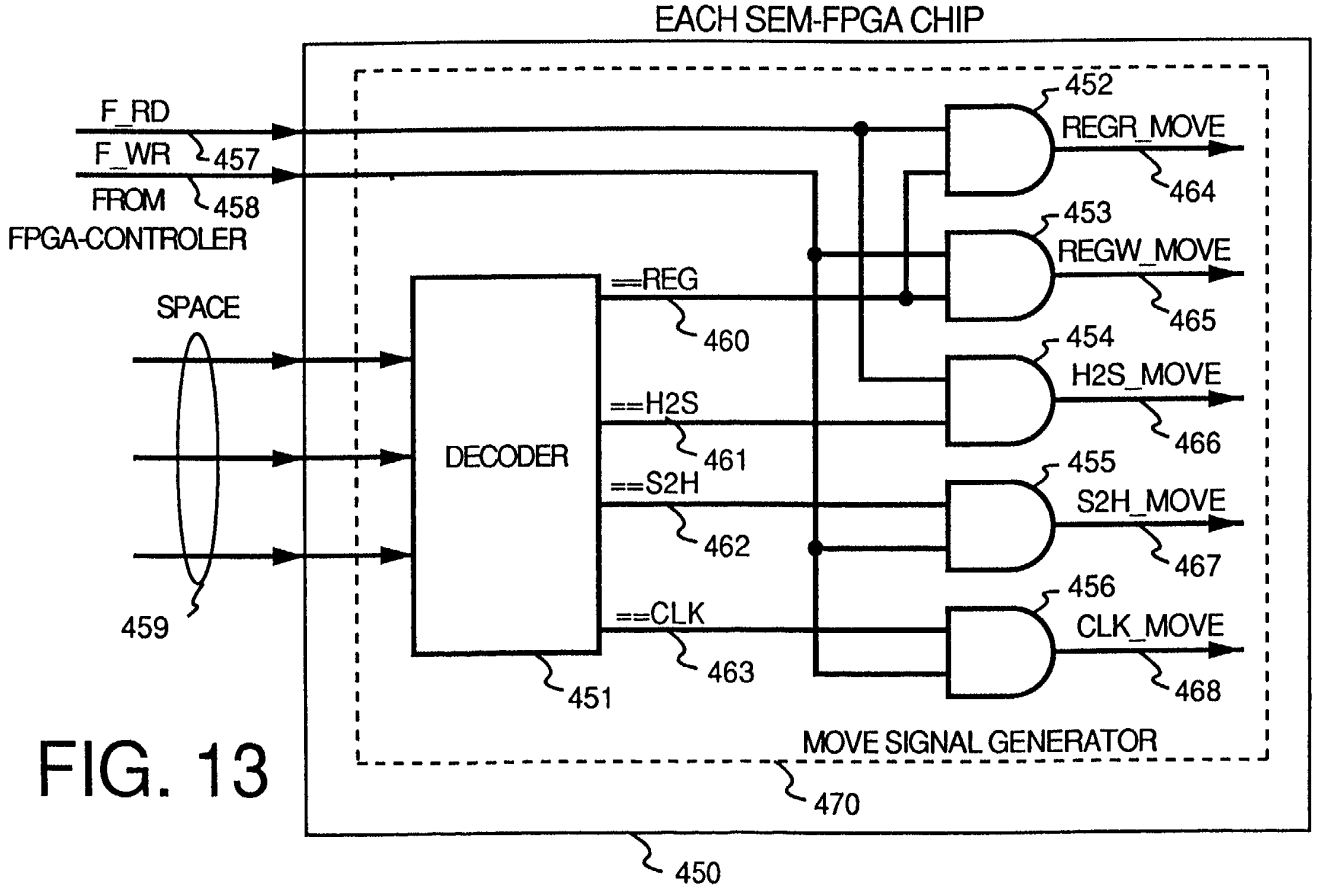

EACH SEM-FPGA CHIP
F_RD
F_WR
457
FROM
458
FPGA-CONTROLER
SPACE
459
DECODER
==REG
460
==H2S
461
==S2H
462
==CLK
463
451
452
REGR_MOVE
464
453
REGW_MOVE
465
454
H2S_MOVE
466
455
S2H_MOVE
467
456
CLK_MOVE
468
MOVE SIGNAL GENERATOR
470
450

FIG. 13

TOOTOT "h2T00660

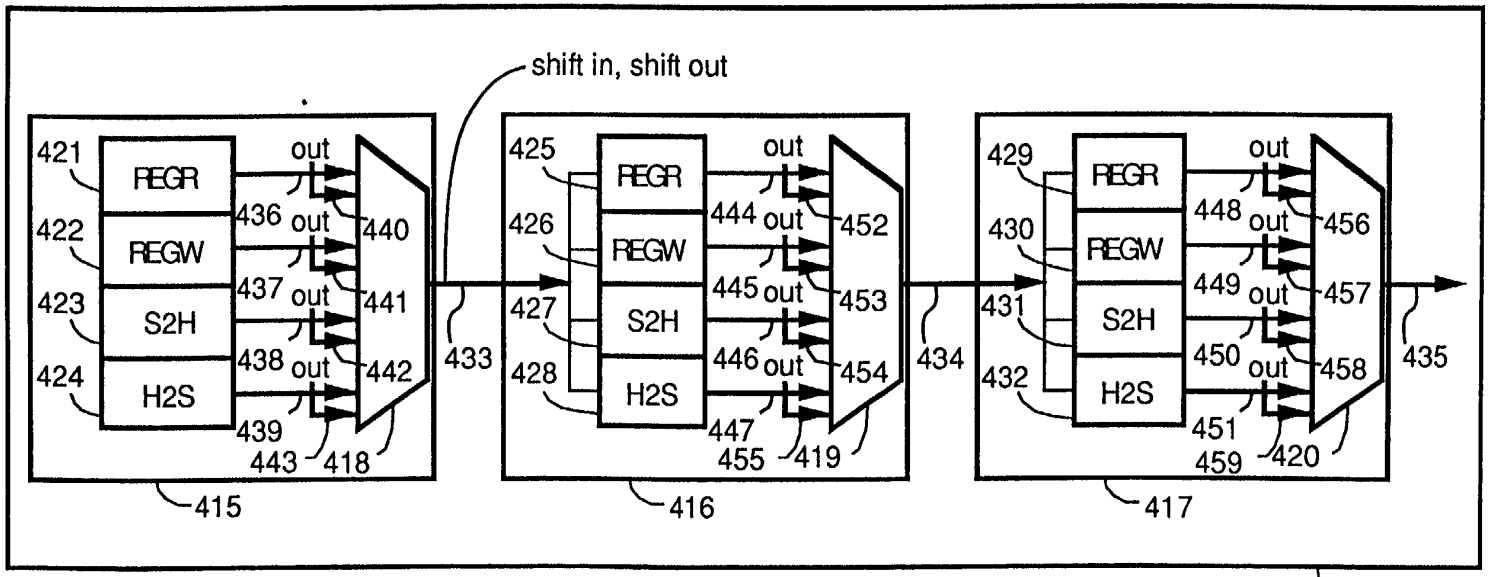


FIG. 14

T00T0T"h2T00660

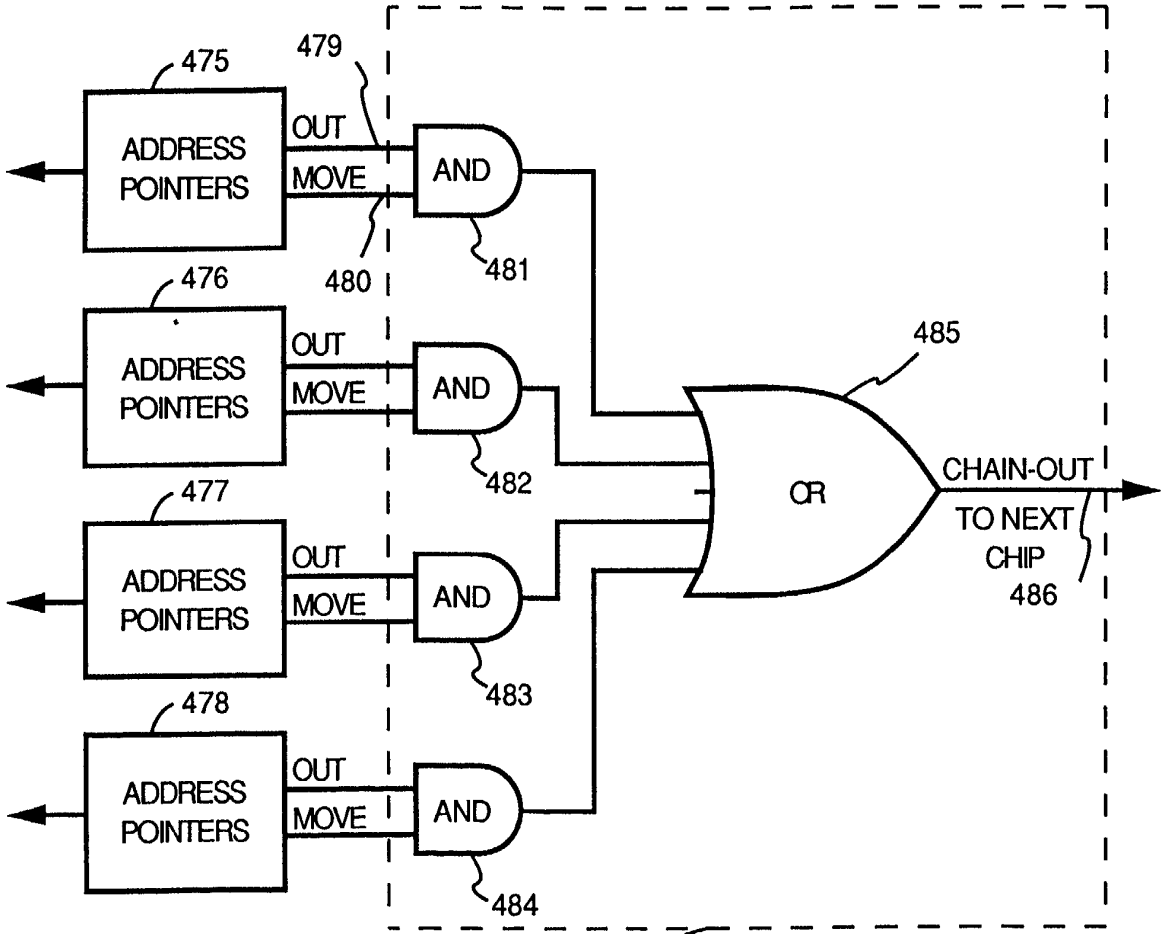


FIG. 15

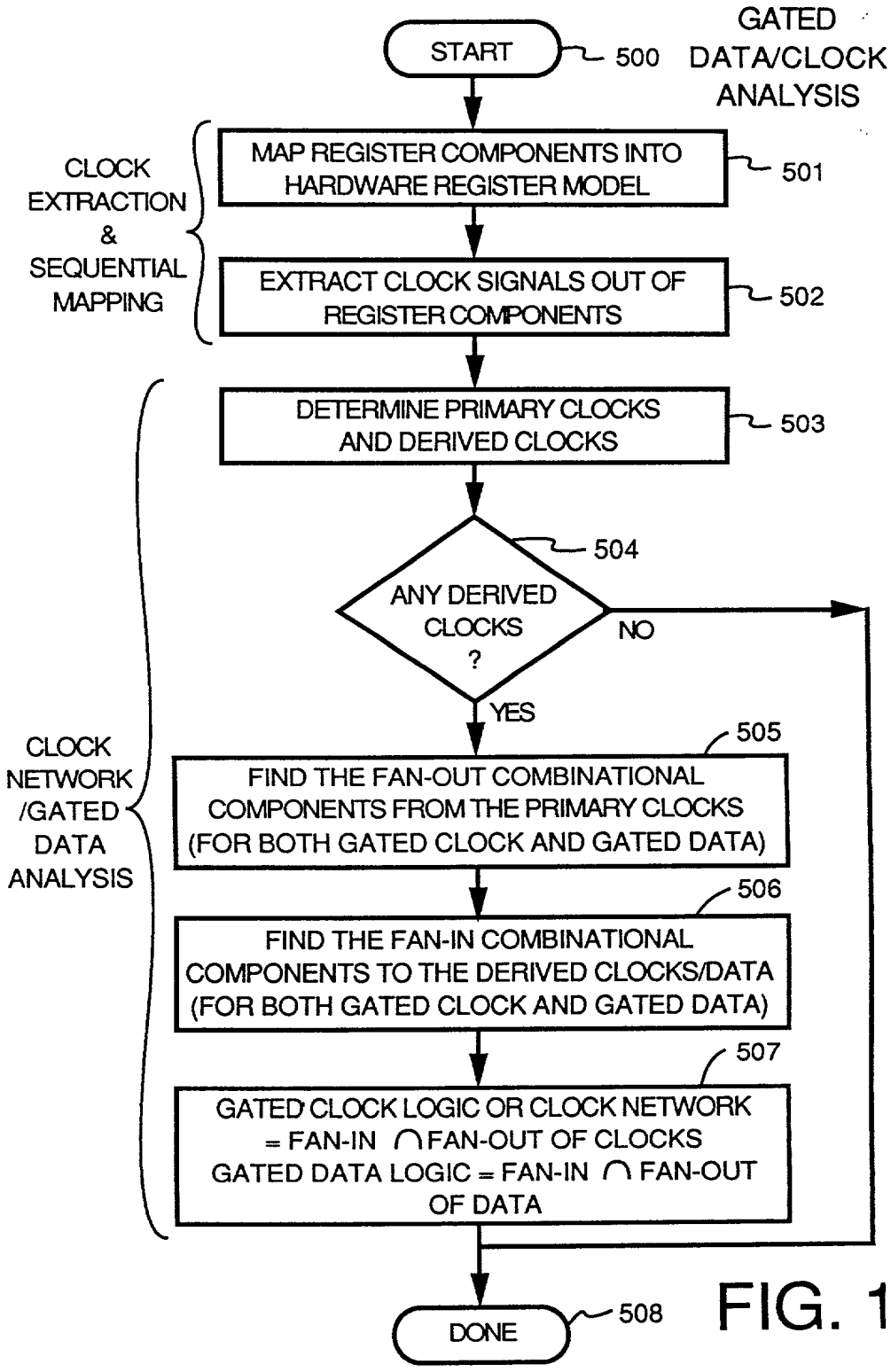

GATED DATA/CLOCK ANALYSIS
START
500
CLOCK EXTRACTION & SEQUENTIAL MAPPING
MAP REGISTER COMPONENTS INTO HARDWARE REGISTER MODEL
501
EXTRACT CLOCK SIGNALS OUT OF REGISTER COMPONENTS
502
DETERMINE PRIMARY CLOCKS AND DERIVED CLOCKS
503
504
ANY DERIVED CLOCKS ?
NO
YES
CLOCK NETWORK /GATED DATA ANALYSIS
505
FIND THE FAN-OUT COMBINATIONAL COMPONENTS FROM THE PRIMARY CLOCKS (FOR BOTH GATED CLOCK AND GATED DATA)
506
FIND THE FAN-IN COMBINATIONAL COMPONENTS TO THE DERIVED CLOCKS/DATA (FOR BOTH GATED CLOCK AND GATED DATA)
507
GATED CLOCK LOGIC OR CLOCK NETWORK = FAN-IN ∩ FAN-OUT OF CLOCKS
GATED DATA LOGIC = FAN-IN ∩ FAN-OUT OF DATA
DONE
508

FIG. 16

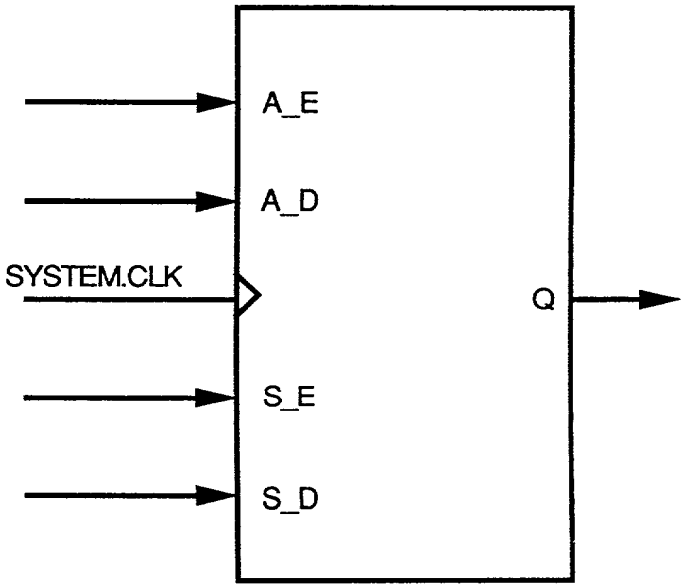


FIG. 17

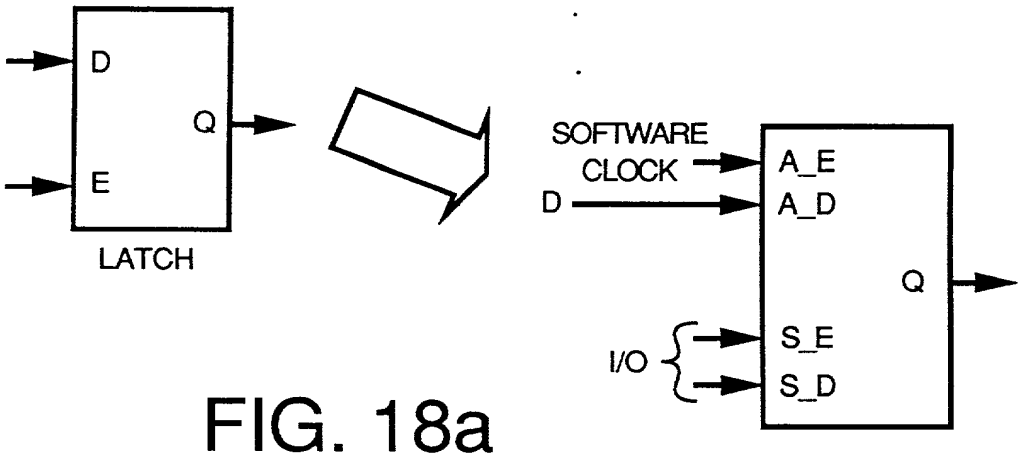


FIG. 18a

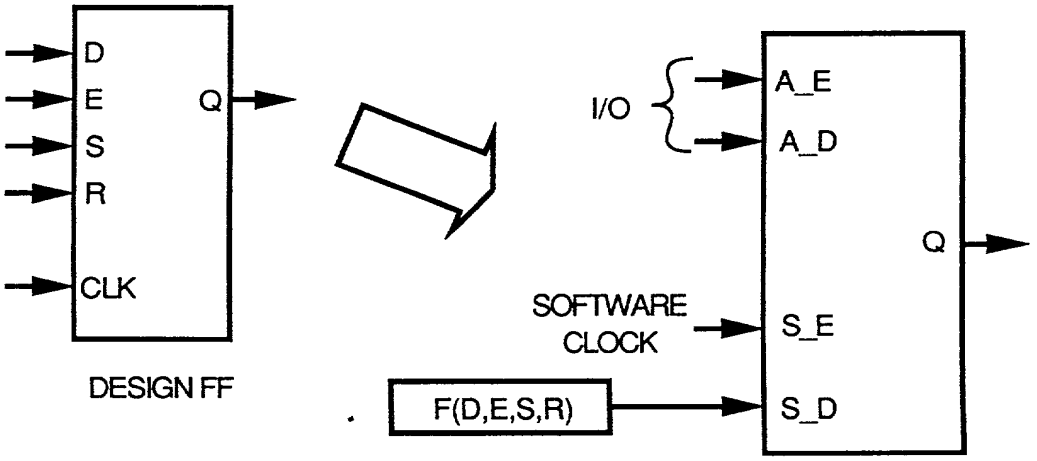


FIG. 18b

T00T0T*h2T00660

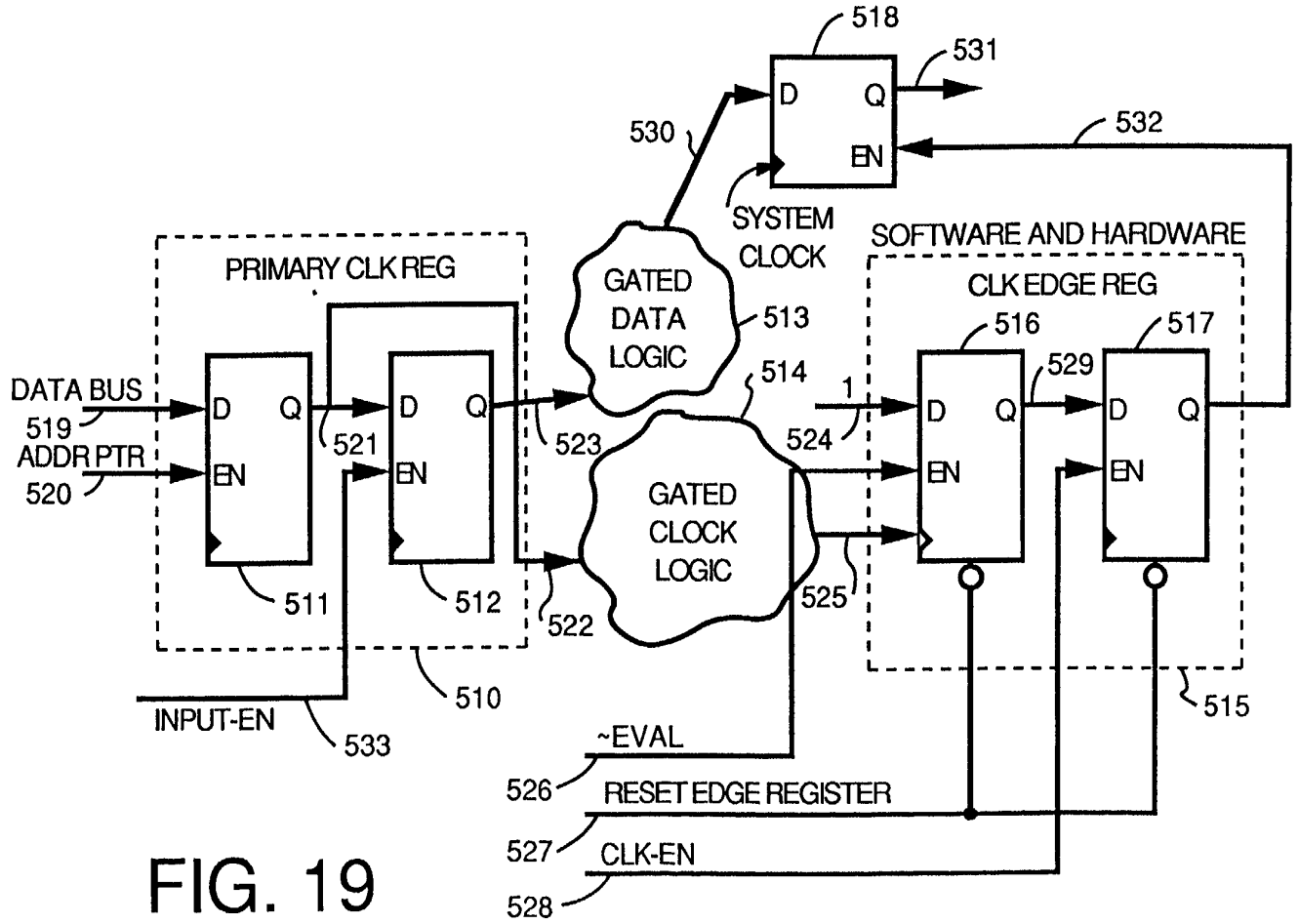


FIG. 19

TOOTOT"h2TOD660

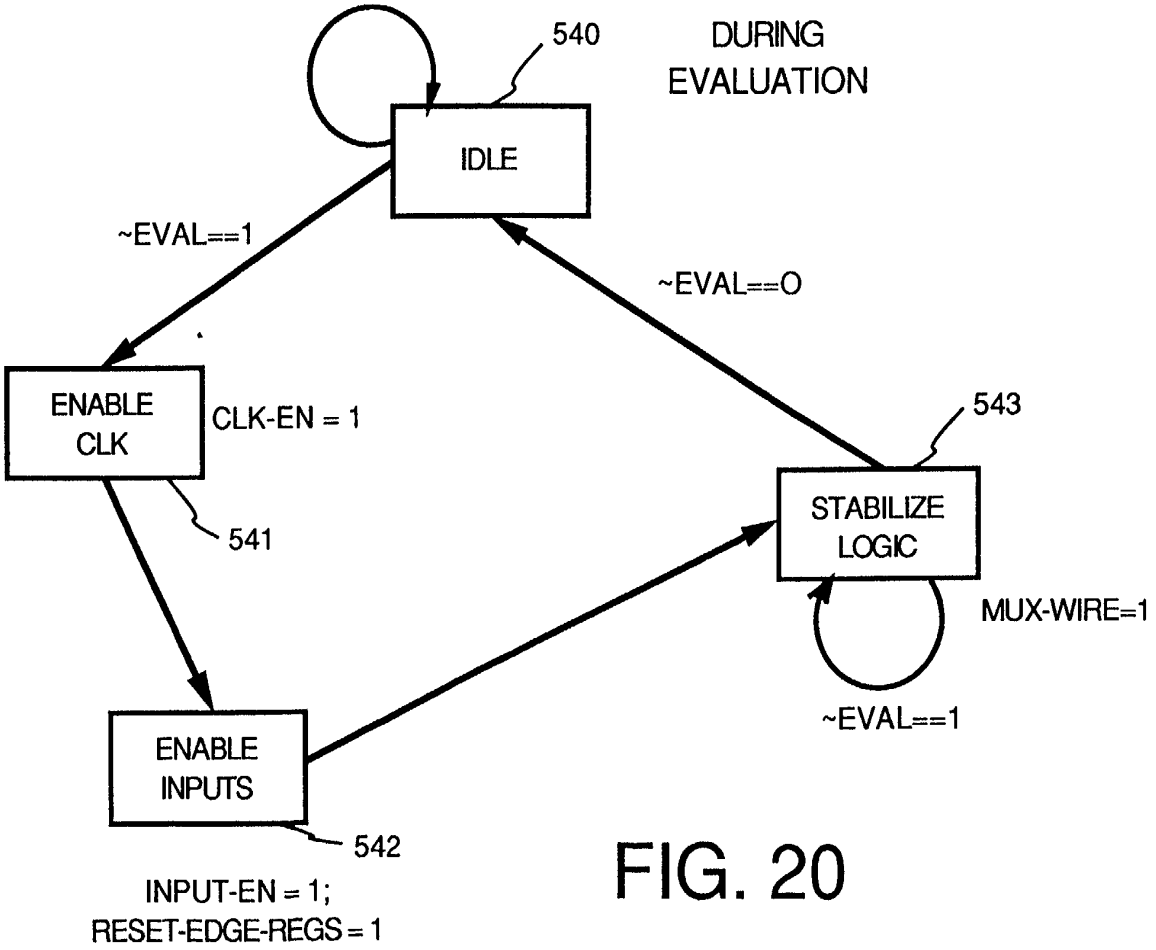


FIG. 20

Proposed

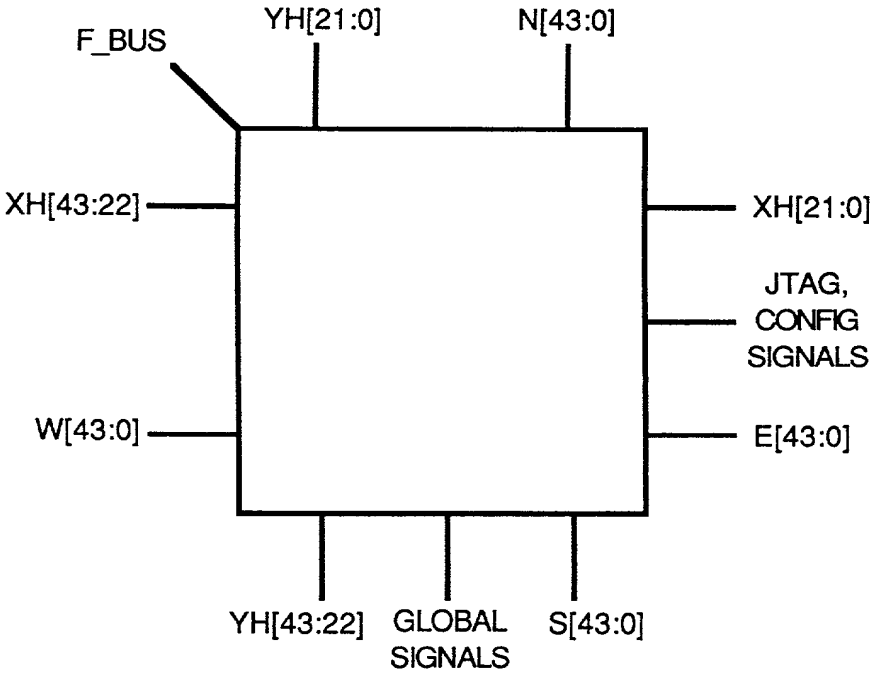


FIG. 21

FOOTOT "+27006660

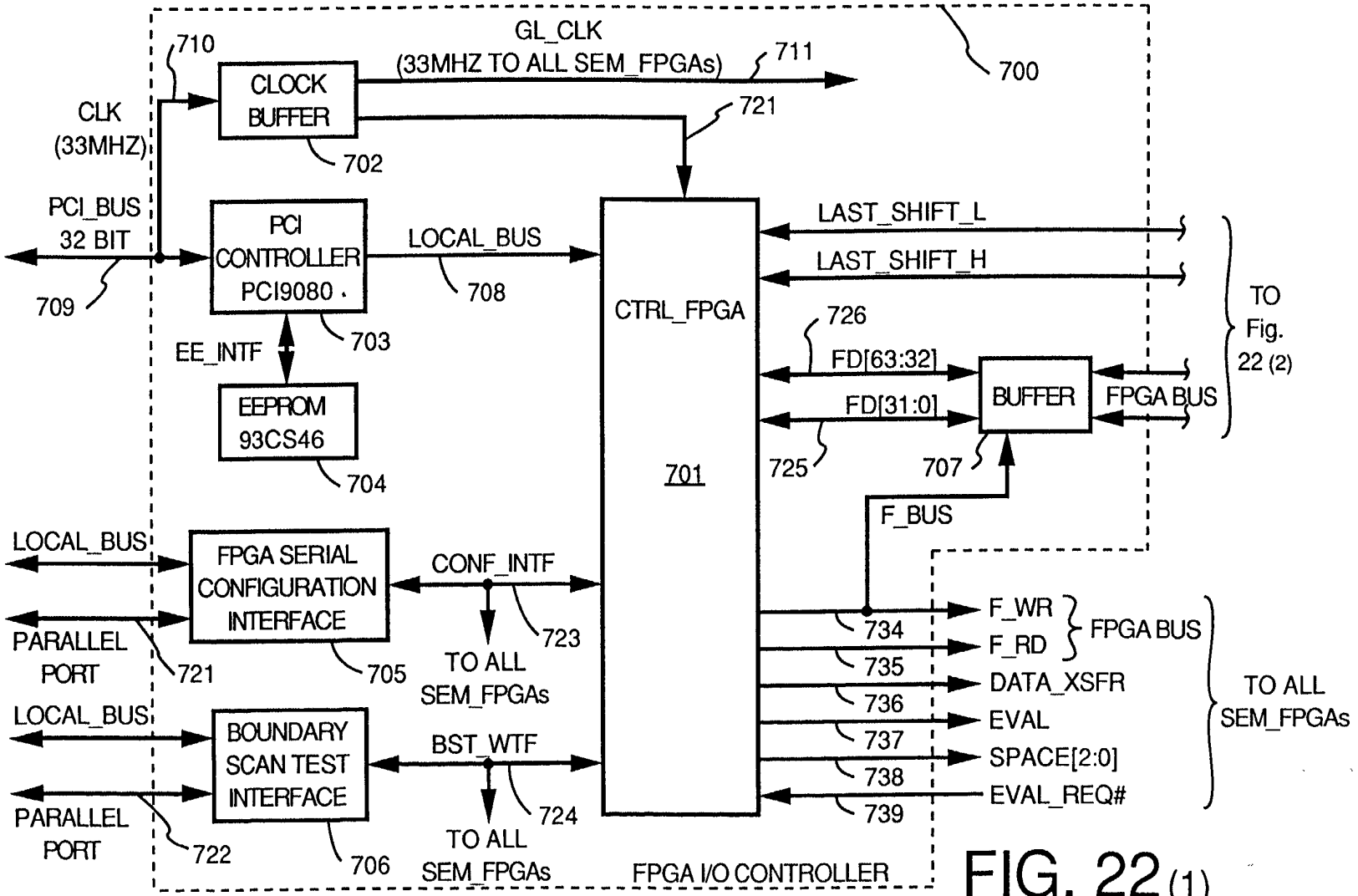


FIG. 22 (1)

FOOTOF "h2100660

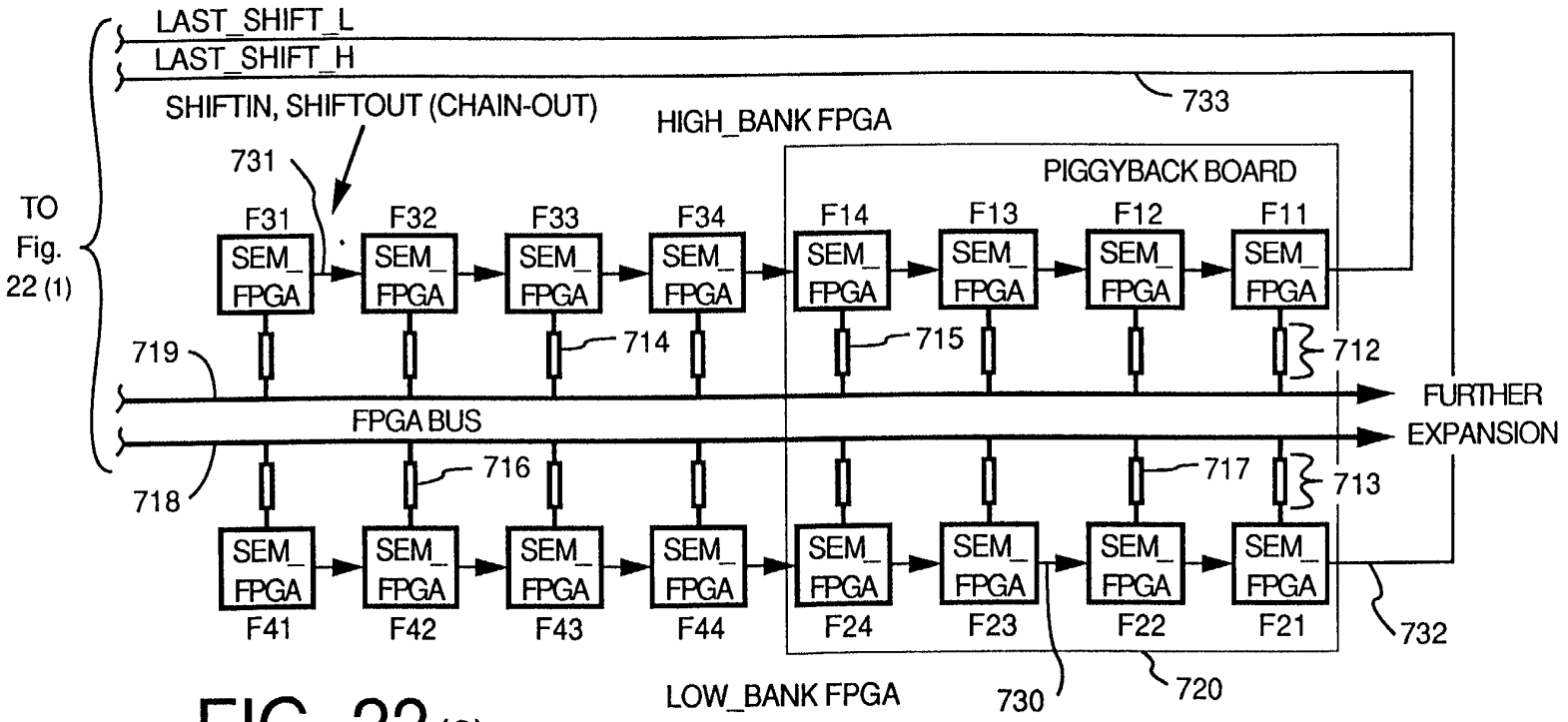


FIG. 22 (2)

TOOTOT "h2T00660

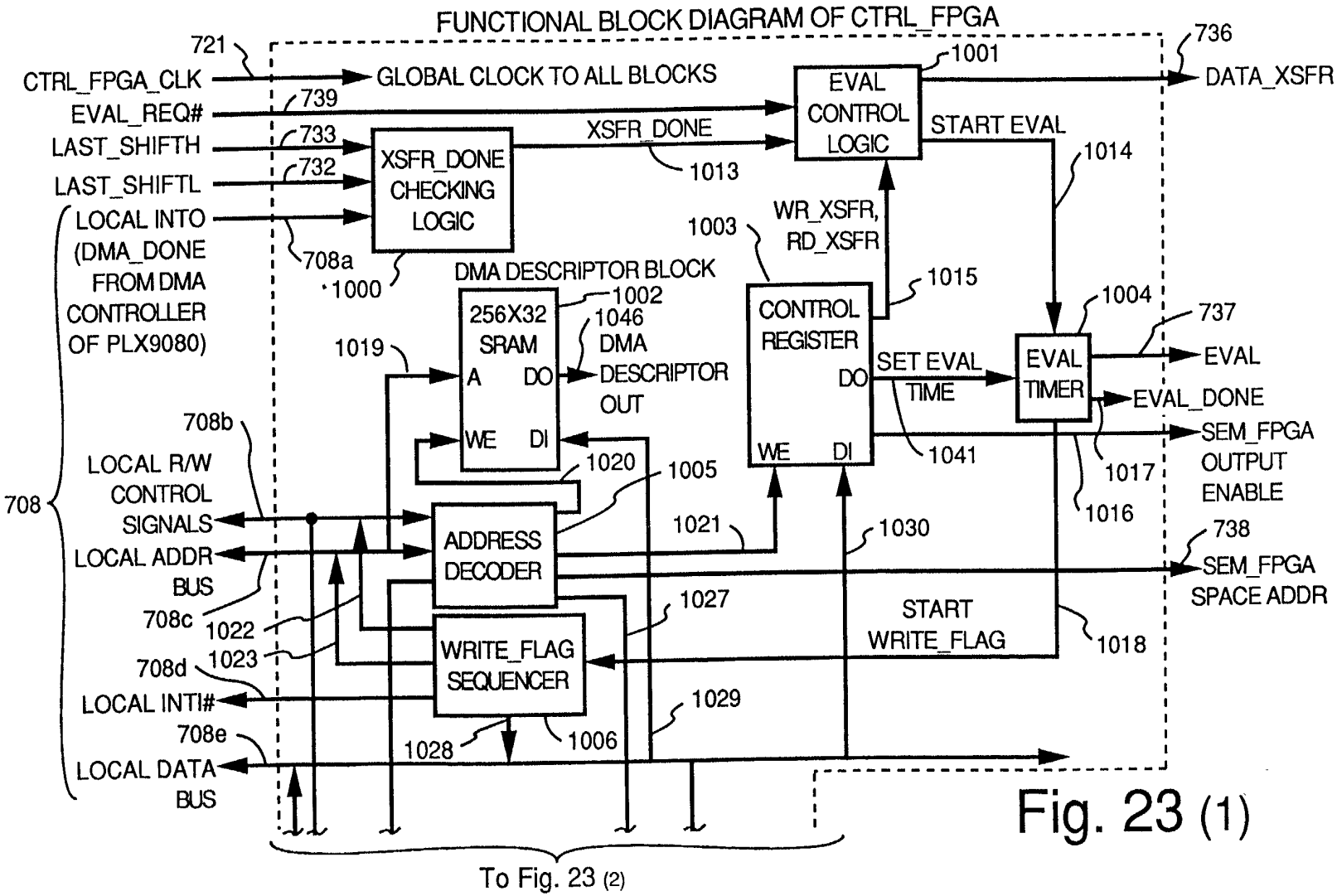


Fig. 23 (1)

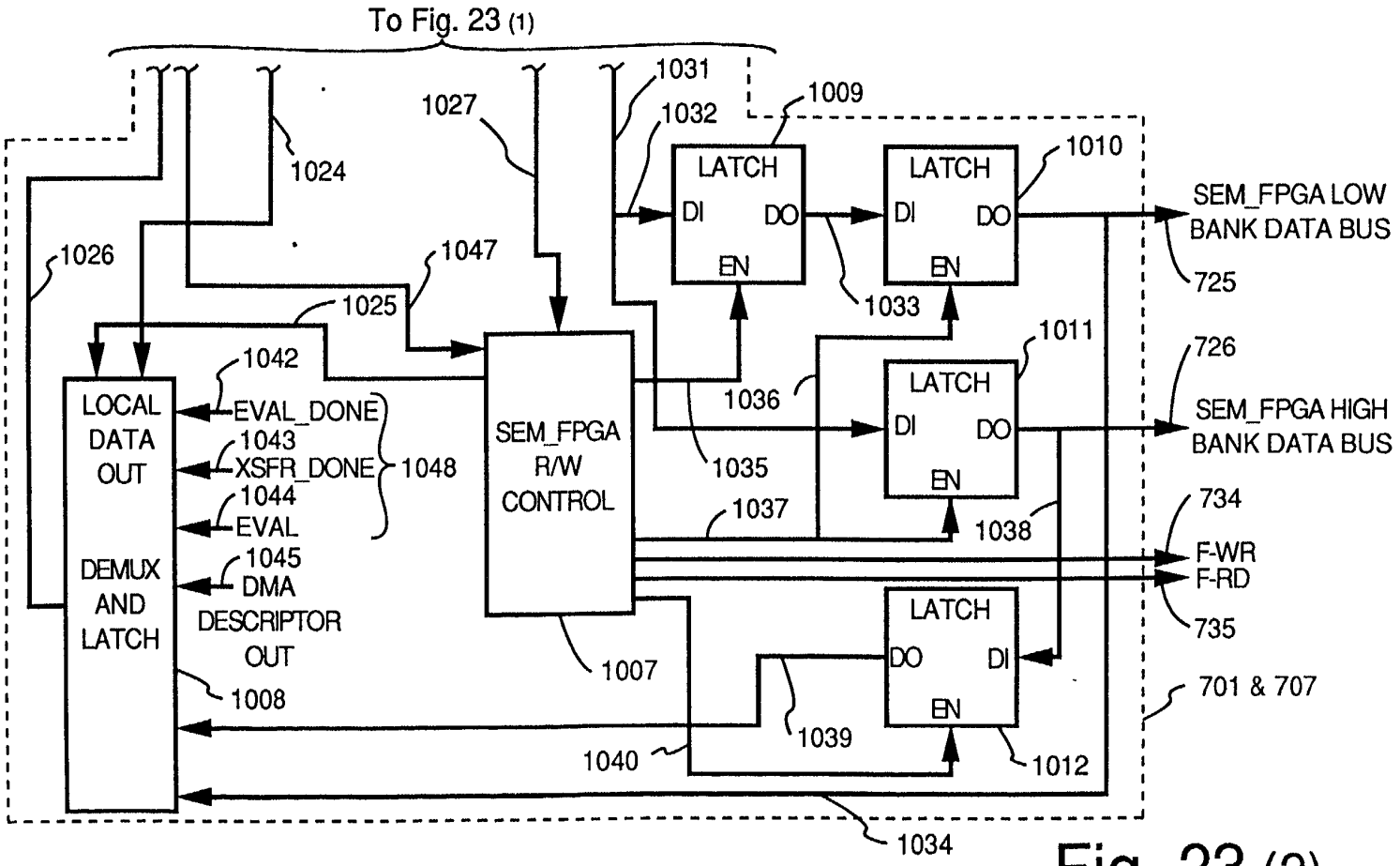

To Fig. 23 (1)
1024
1027
1031
1032
1009
1010
LATCH
DI
DO
EN
SEM_FPGA LOW
BANK DATA BUS
725
1026
1047
1025
1033
1011
726
1042
LOCAL
DATA
OUT
EVAL_DONE
1043
XSFR_DONE
1048
1044
EVAL
1045
DMA
DESCRIPTOR
OUT
1036
SEM_FPGA
R/W
CONTROL
1035
1037
1038
SEM_FPGA HIGH
BANK DATA BUS
734
F-WR
F-RD
735
DEMUX
AND
LATCH
1008
1007
1039
1040
1012
701 & 707
1034

Fig. 23 (2)

TOOTOT "H2T00660

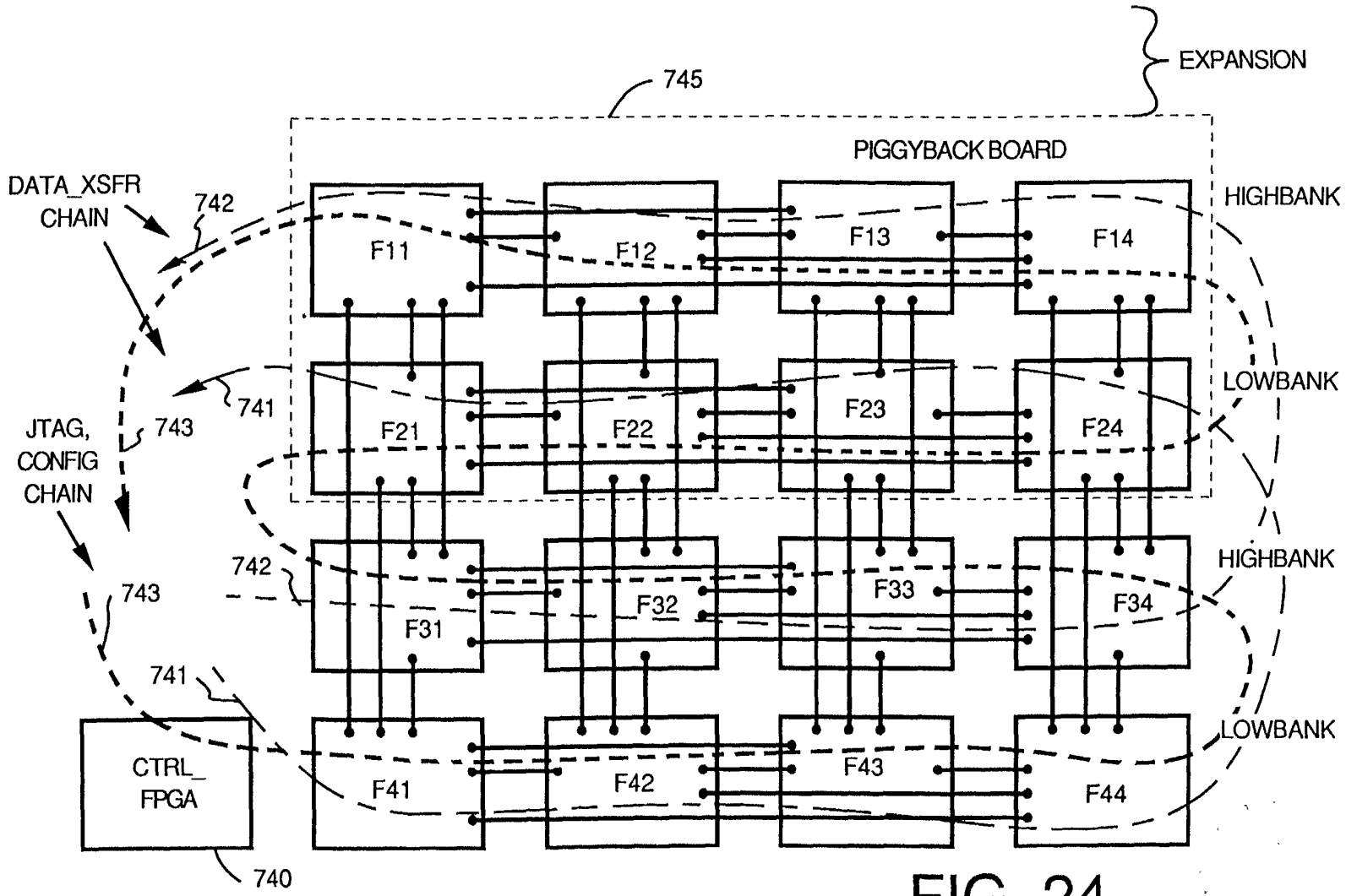


FIG. 24

HARDWARE START-UP

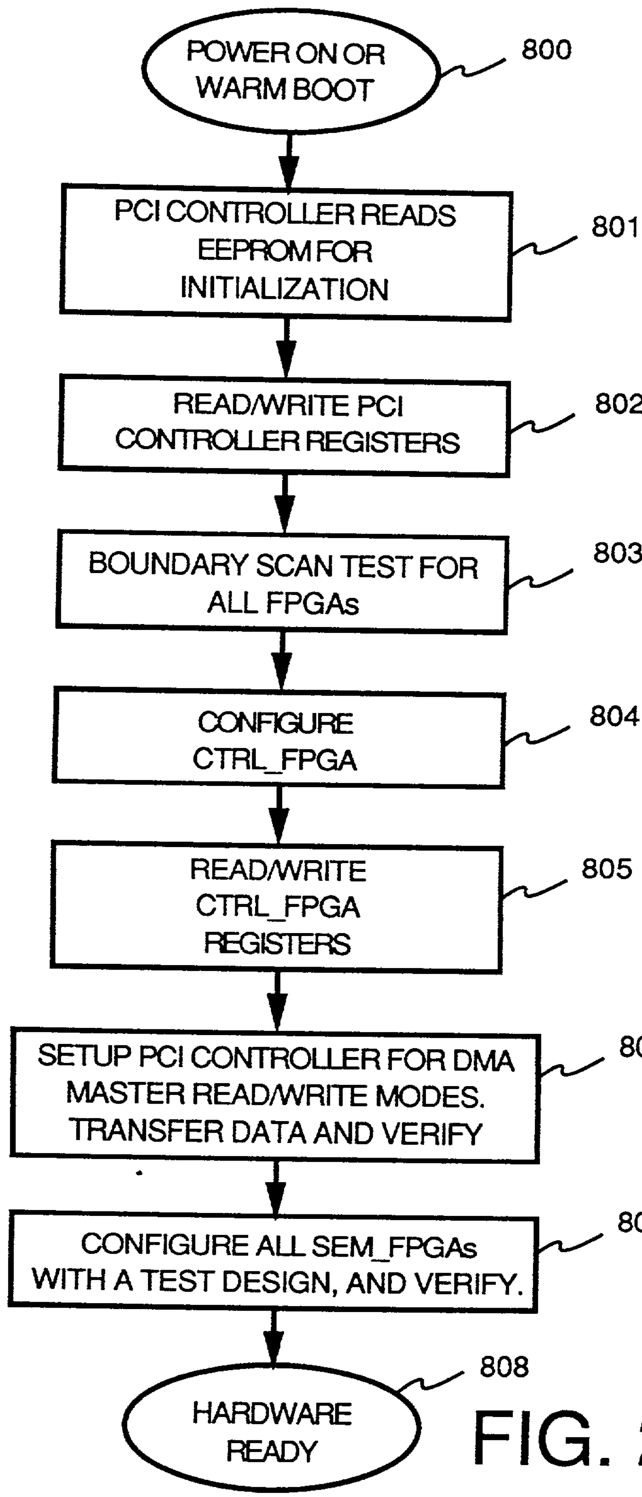


FIG. 25

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(posedge clock or negedge reset)
   if(~reset)
      q = 0;
   else
      q = d;
endmodule

module example;
   wire d1, d2, d3;
   wire q1, q2, q3;
   reg sigin;
   wire sigout;
   reg clk, reset;

   register reg1 (clk, reset, d1, q1);
   register reg2 (clk, reset, d2, q2);
   register reg3 (clk, reset, d3, q3);

   assign d1 = sigin ^ q3;
   assign d2 = q1 ^ q3;
   assign d3 = q2 ^ q3;
   assign sigout = q3;

   // a clock generator
   always
   begin
      clk = 0;
      #5;
      clk = 1;
      #5;
   end

   // a signal generator
   always
   begin
      #10;
      sigin = $random;
   end

   // initialization
   initial
   begin
      reset = 0;
      sigin = 0;
      #1;
      reset =1;
      #5;
      $monitor($time, " %b, %b", sigin, sigout);
      #1000 $finish;
   end
   end module
```

Fig. 26

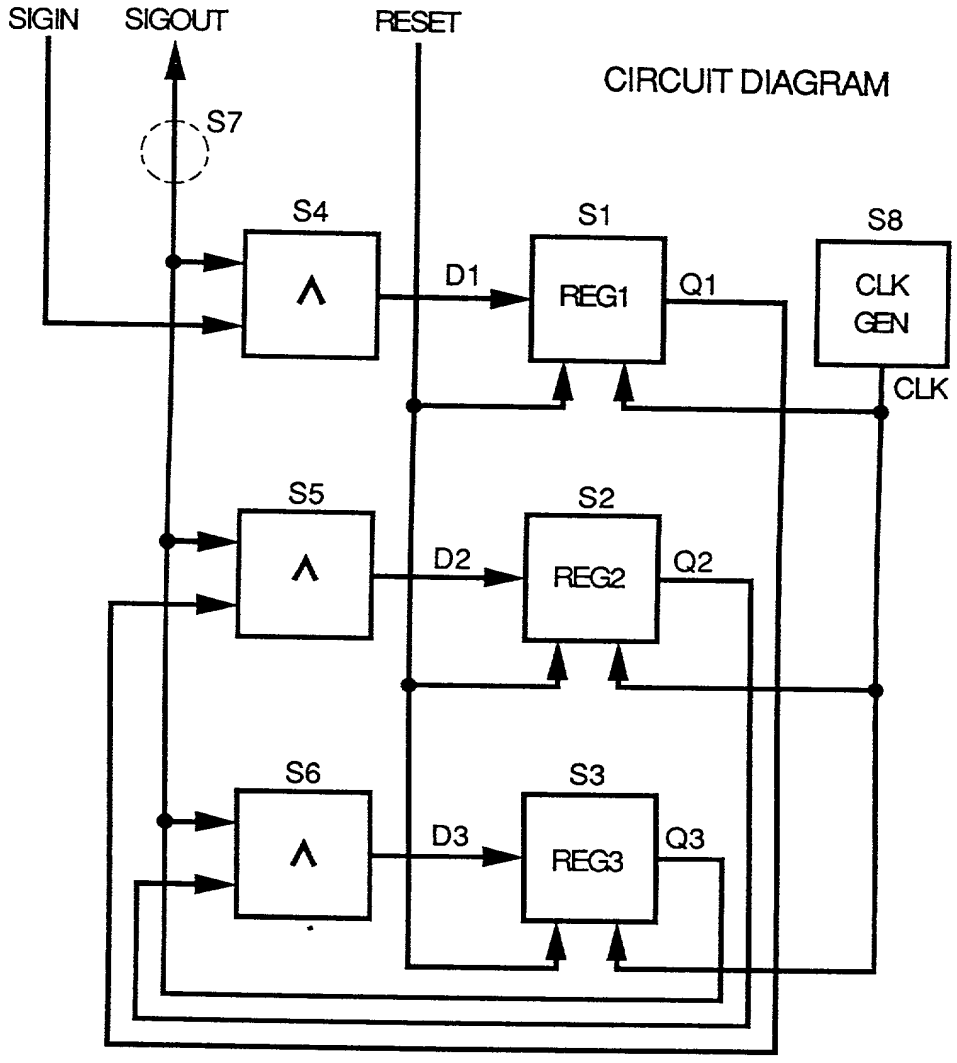

SIGIN
SIGOUT
RESET
CIRCUIT DIAGRAM
S7
S4
S1
S8
D1
REG1
Q1
CLK
GEN
CLK
S5
S2
D2
REG2
Q2
S6
S3
D3
REG3
Q3

FIG. 27

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(post edge clock or negedge reset)
   if(~reset)
     q = 0
   else
     q = d;

endmodule
```

Register definition

900

```
module example;
   wire d1, d2, d3;
   ware q1, q2, q3;
```

Wire interconnection info

907

```
   reg sigin;
   wire sigout;
   reg clk, reset;
```

Test-bench input -- 908

Test-bench output -- 909

```
S1  register reg 1 (clk, reset, d1, q1);
S2  register reg 2 (clk, reset, d2, q2);
S3  register reg 3 (clk, reset, d3, q3);
```

Register component

901

```
S4  assign d1 = sigin ^ q3;
S5  assign d2 = q1 ^ 3;
S6  assign d3 = q2 ^ q3;
S7  assign signout = q3;
```

Combinational component

902

```
S8  // a clock generator
    always
    begin
      clk = 0;
      #5;
      clk = 1;
      #5;
    end
```

Clock component

903

```
S9  // a signal generator
    always
    begin
      #10;
      sigin = $random;
    end
```

Test-bench component (Driver)

904

Fig. 28

```
    // initialization
    initial
S10 begin
      reset = 0;
      sigin = 0;
      #1;
S11   reset = 1;
      #5;
S12   $monitor($time, "%b, %b", sigin, sigout);
      #1000 $finish;
    end
    end module
```

Test-bench component (initialization)

905

Test-bench component (monitor)

906

TOSTOT"h2T00660

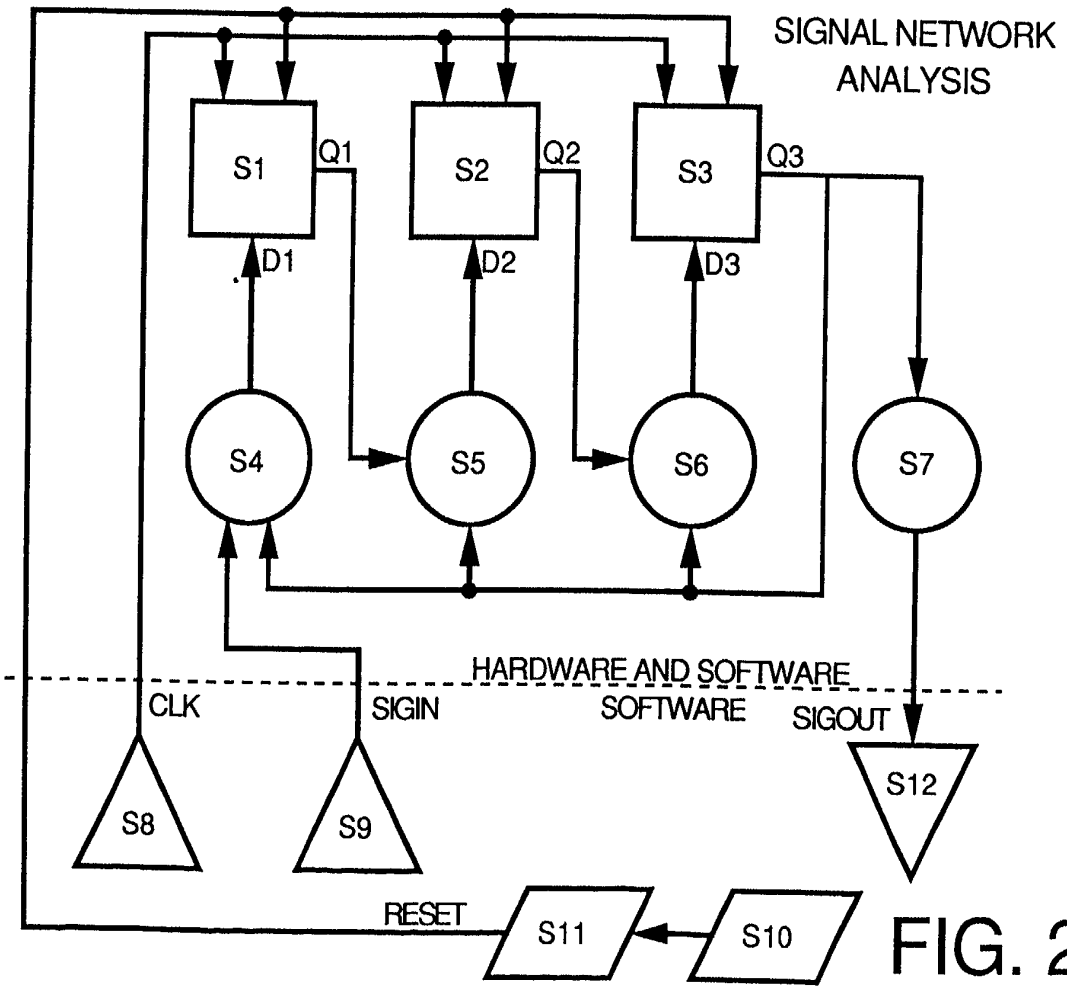


FIG. 29

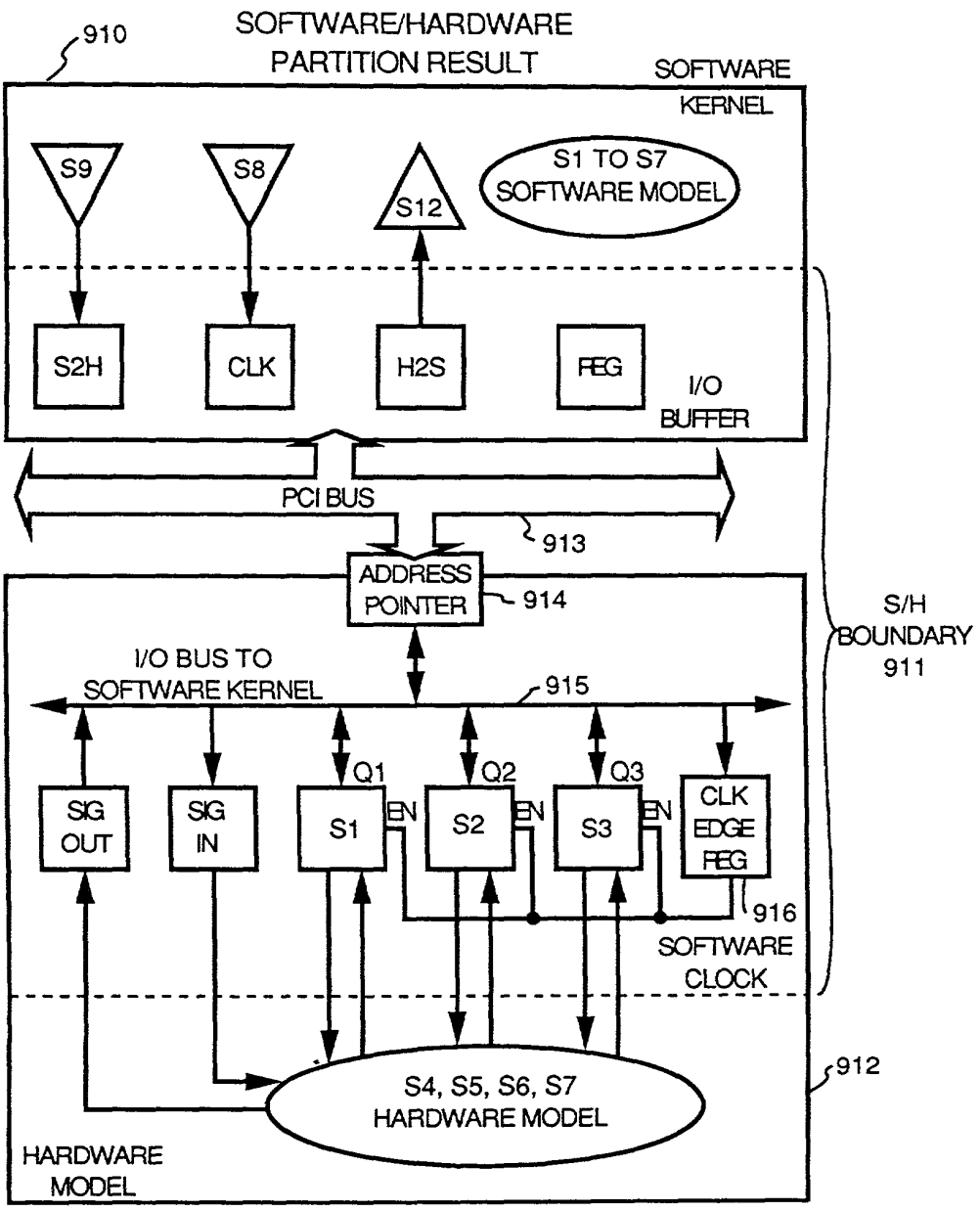

910
SOFTWARE/HARDWARE
PARTITION RESULT
SOFTWARE
KERNEL
S9
S8
S12
S1 TO S7
SOFTWARE MODEL
S2H
CLK
H2S
REG
I/O
BUFFER
PCI BUS
913
ADDRESS
POINTER
914
S/H
BOUNDARY
911
I/O BUS TO
SOFTWARE KERNEL
915
Q1
Q2
Q3
SIG
OUT
SIG
IN
S1
EN
S2
EN
S3
EN
CLK
EDGE
REG
916
SOFTWARE
CLOCK
912
S4, S5, S6, S7
HARDWARE MODEL
HARDWARE
MODEL

FIG. 30

FOITOF "h2F00660

HARDWARE MODEL

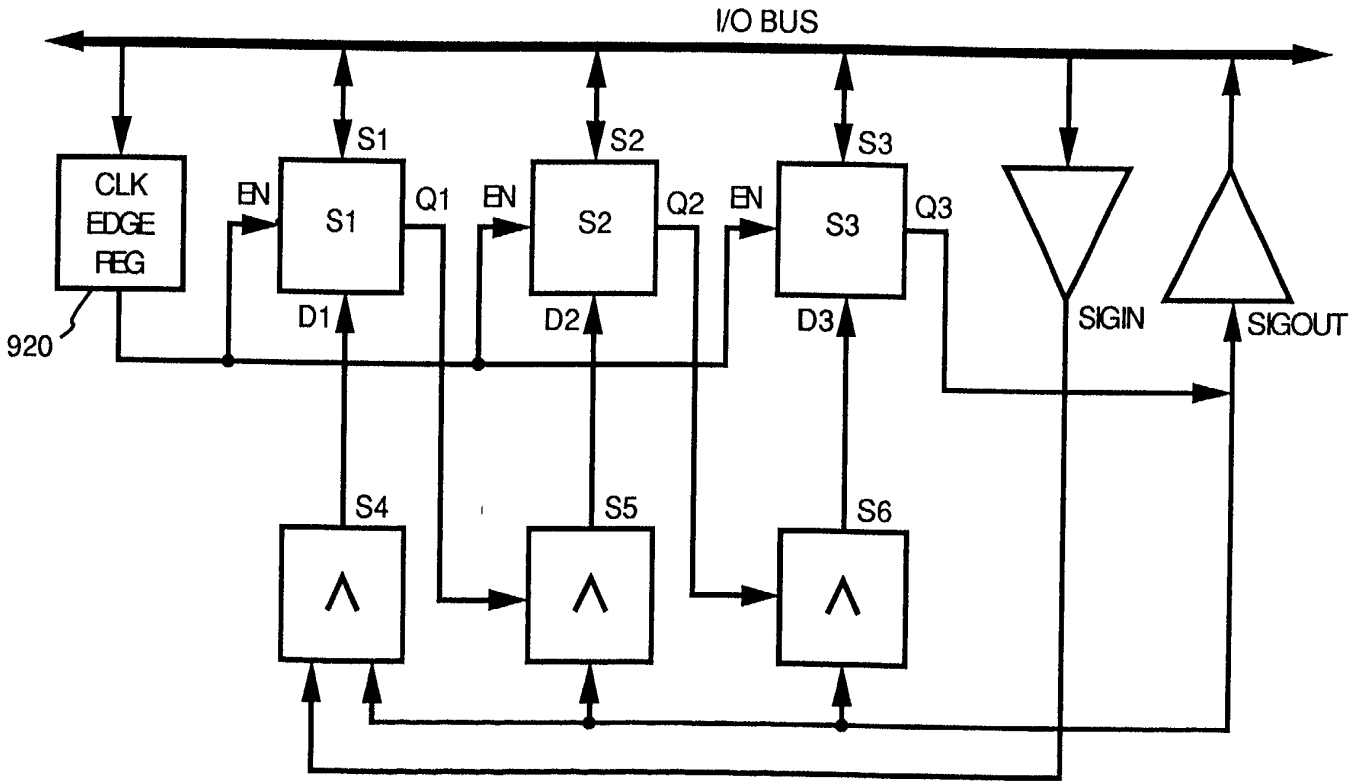
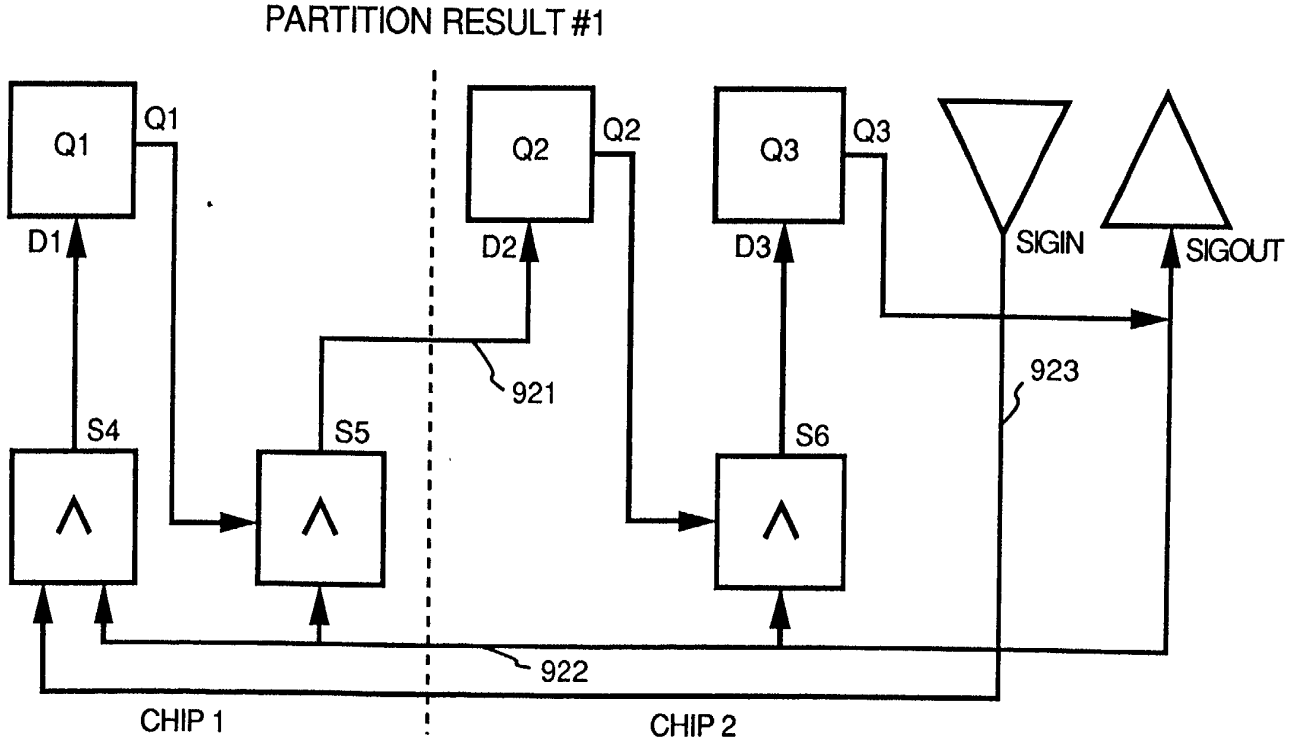


FIG. 31



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 32

FOOTOF"h2F00660

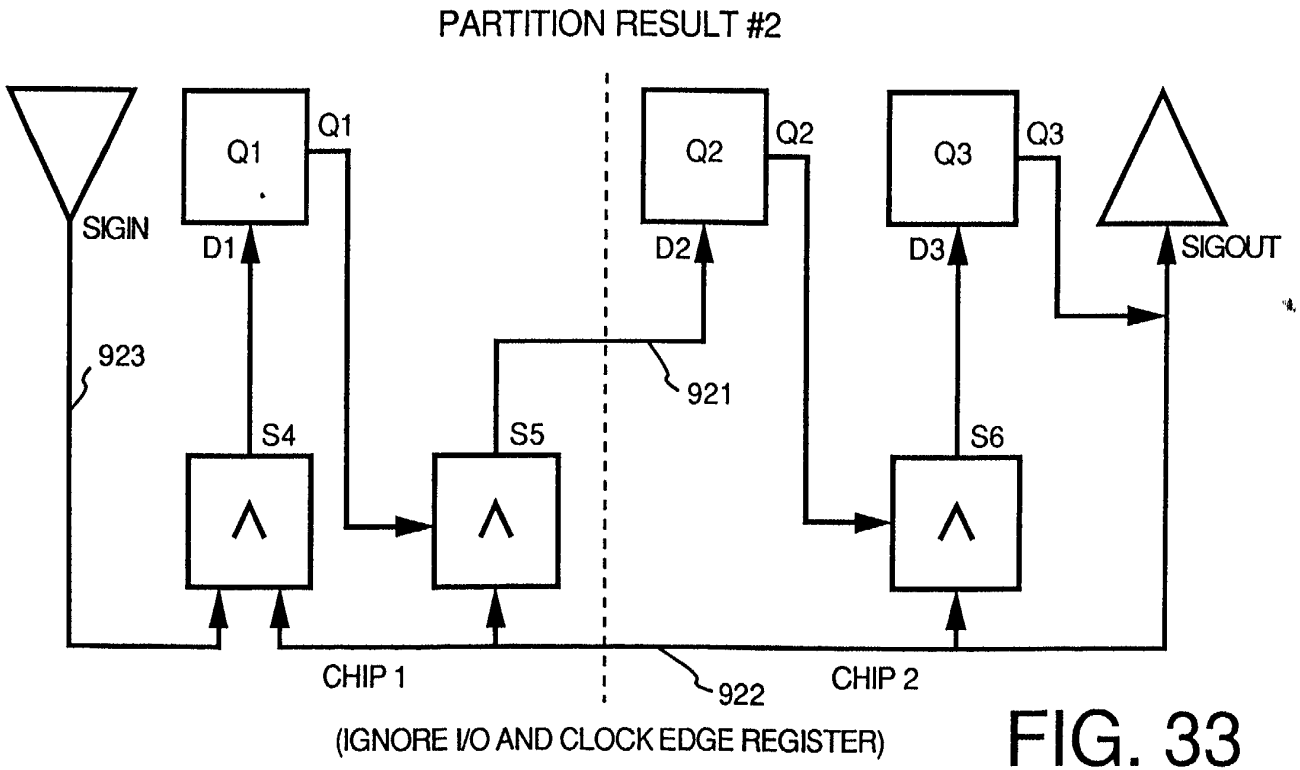


FIG. 33

FOOTOT "h2T00660

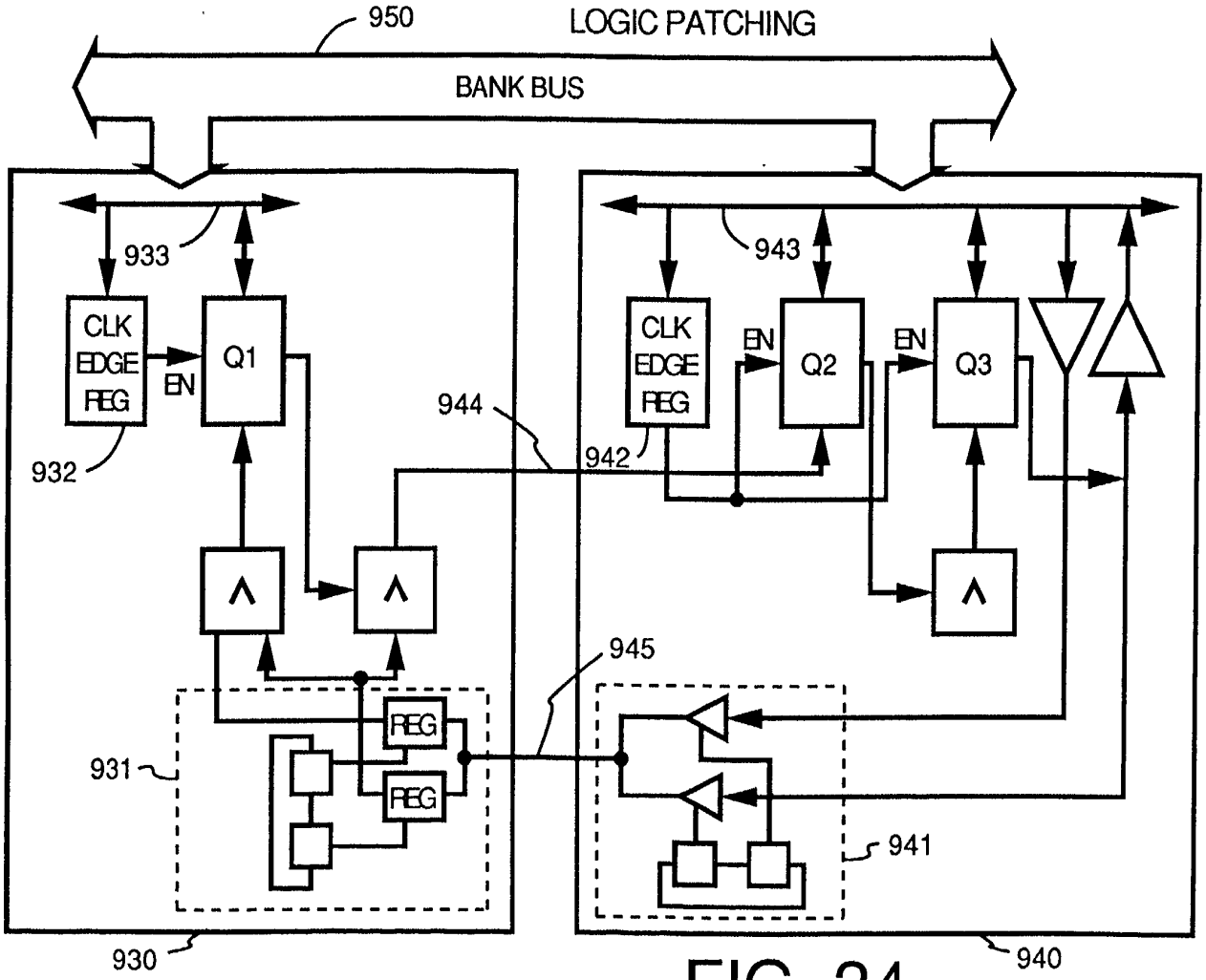


FIG. 34

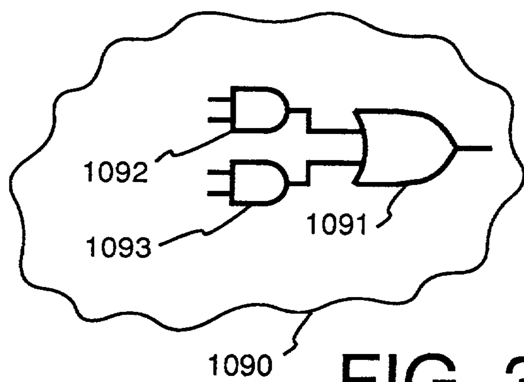


FIG. 35a

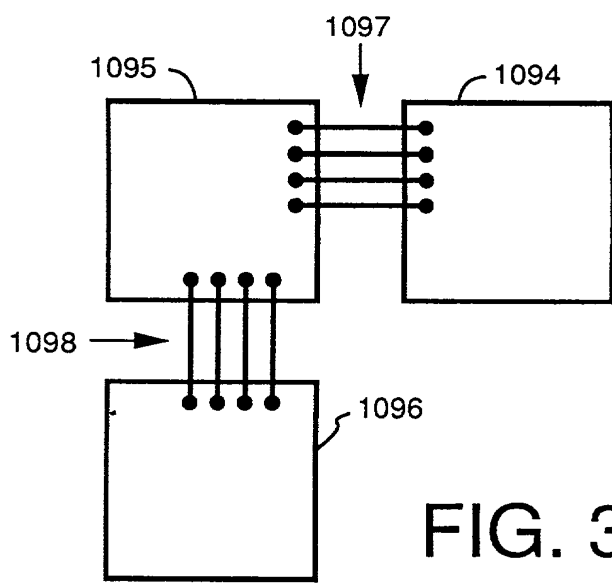


FIG. 35b

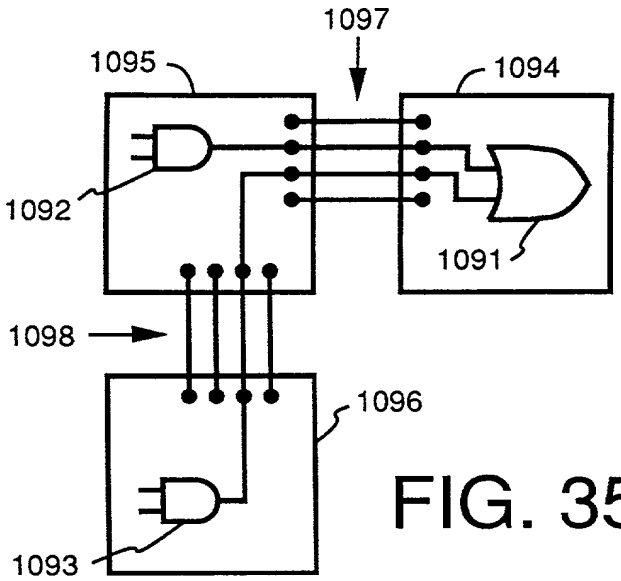


FIG. 35c

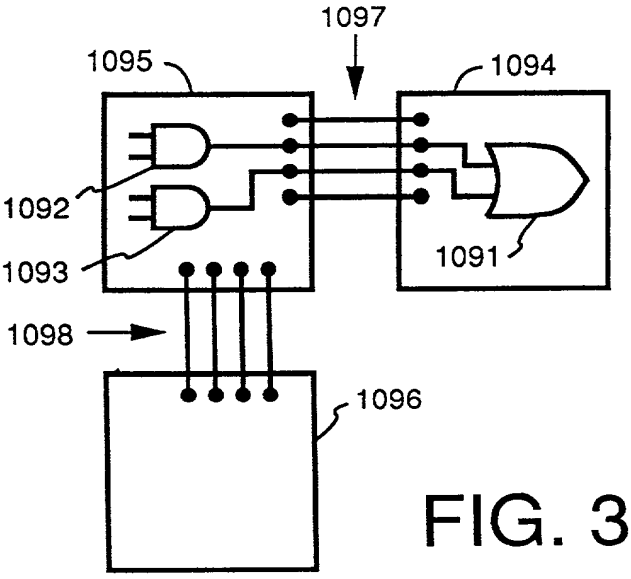
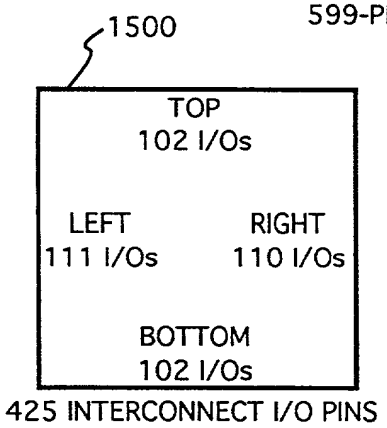


FIG. 35d

I/O PIN OVERVIEW OF
FPGA LOGIC DEVICE

FPGA : 10K130V, 10K250V WITH
599-PIN PGA PACKAGE



45 DEDICATED I/O PINS :

GCLK, FD_BUS[31..0], F_RD, F_WR,
DATAXSFR, SHIFTIN, SHIFTOUT,
SPACE[2..0], EVAL, EV_REQ_N,
DEV_OE, DEV_CLRN

FIG. 36

FPGA INTERCONNECT BUSES

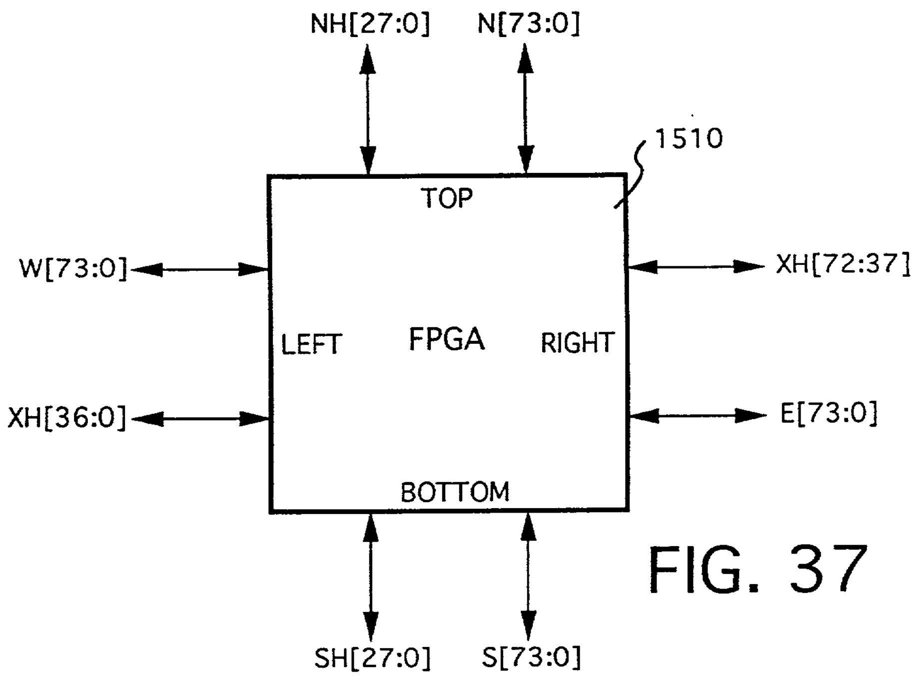


FIG. 37

BOARD CONNECTION - SIDE VIEW

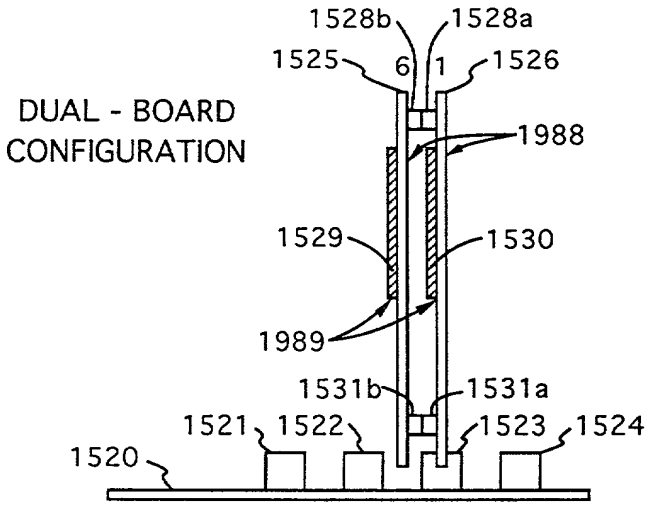


FIG. 38(a)

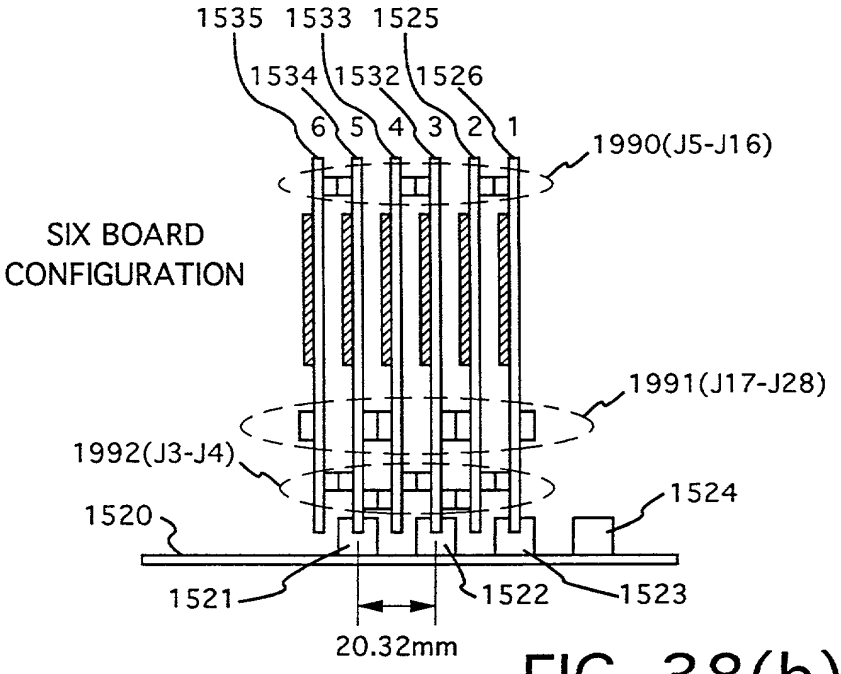


FIG. 38(b)

SIX-BOARD CONFIGURATION DIRECT-NEIGHBOR AND
ONE-HOP FPGA ARRAY-X TORUS, Y MESH

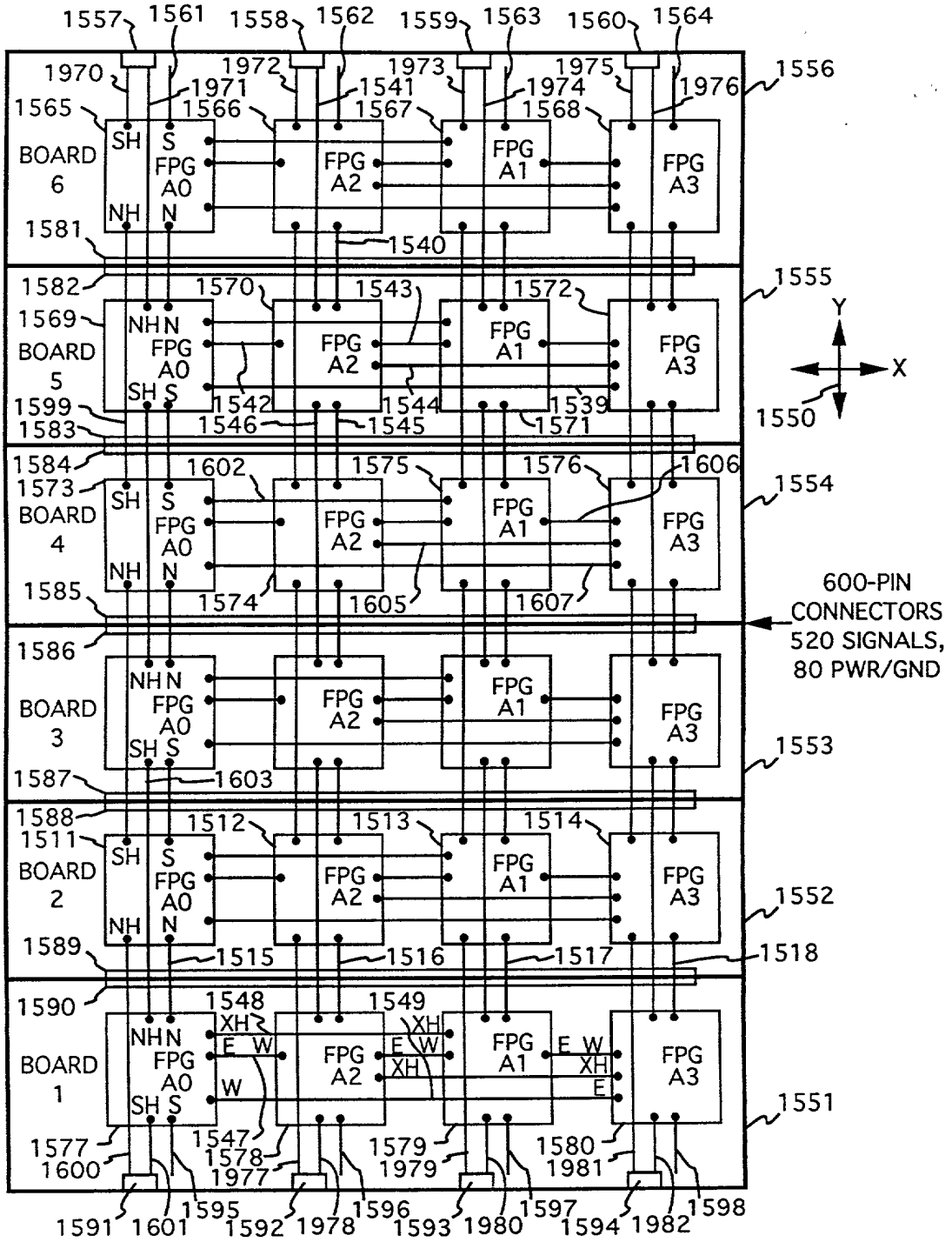


FIG. 39

FPGA ARRAY CONNECTION BETWEEN BOARDS

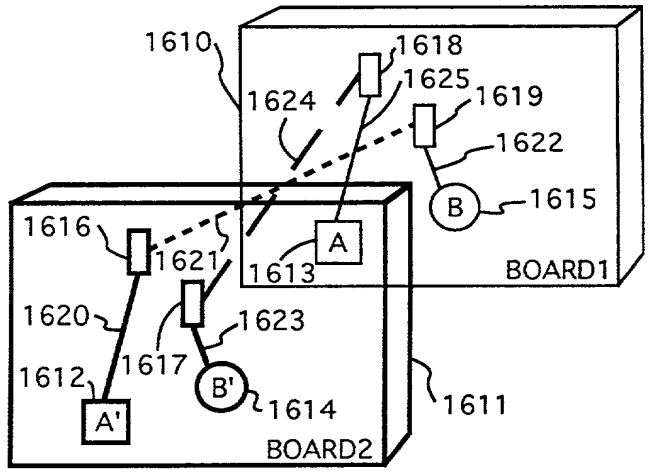


FIG. 40(a)

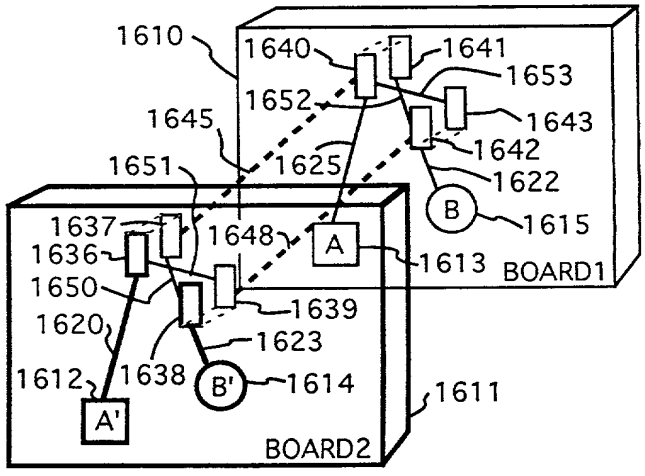


FIG. 40(b)

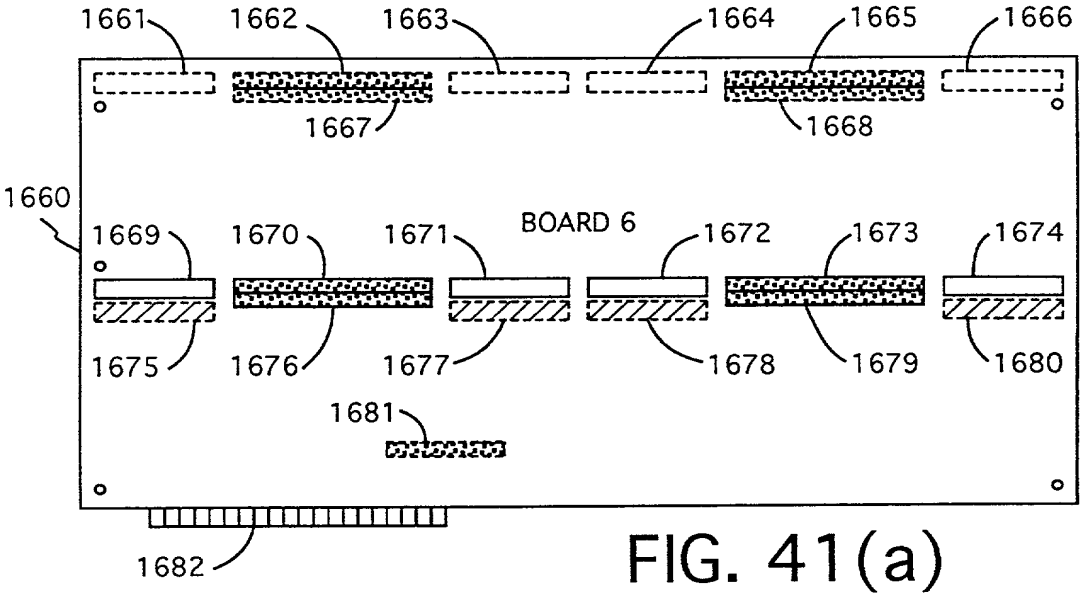


FIG. 41(a)

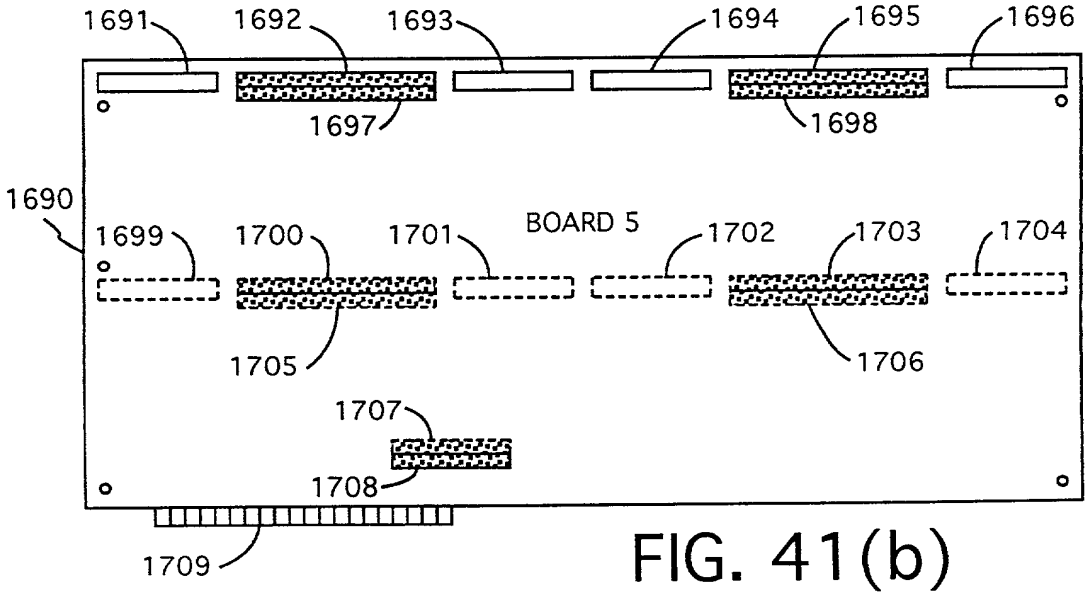


FIG. 41(b)

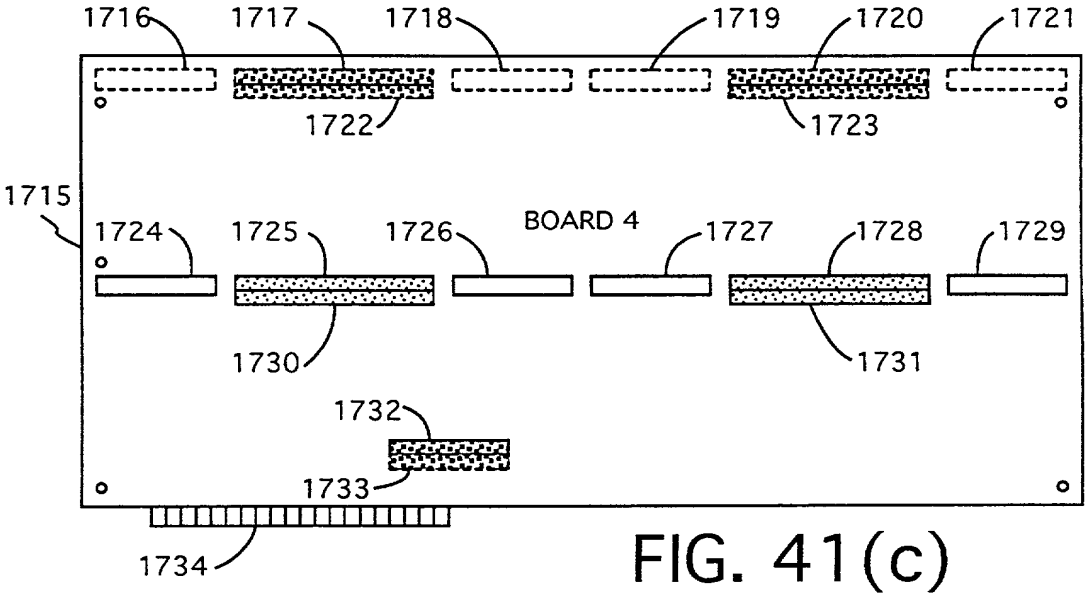


FIG. 41(c)

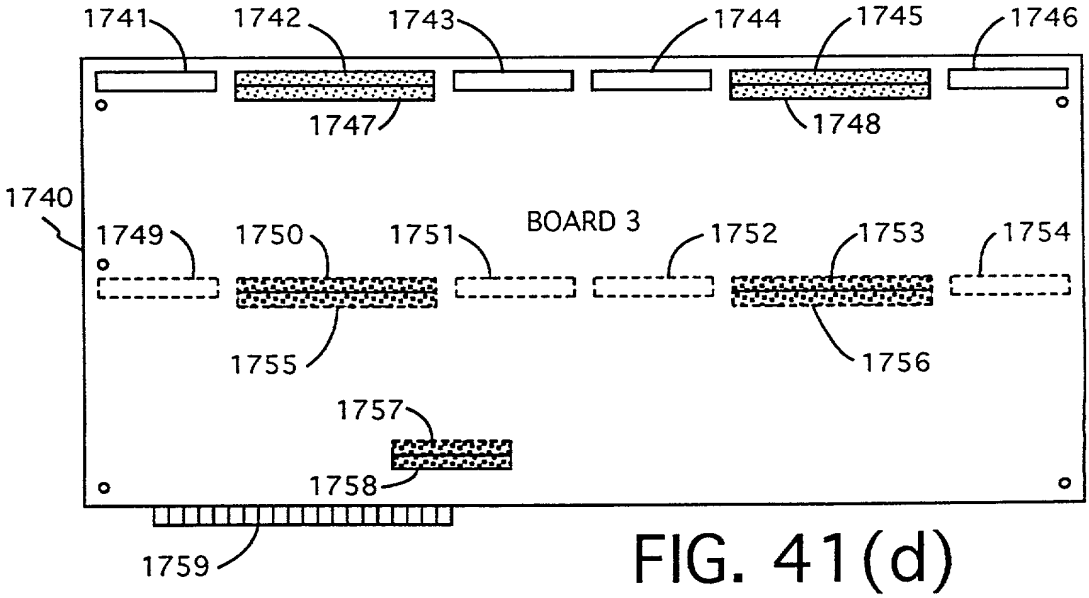


FIG. 41(d)

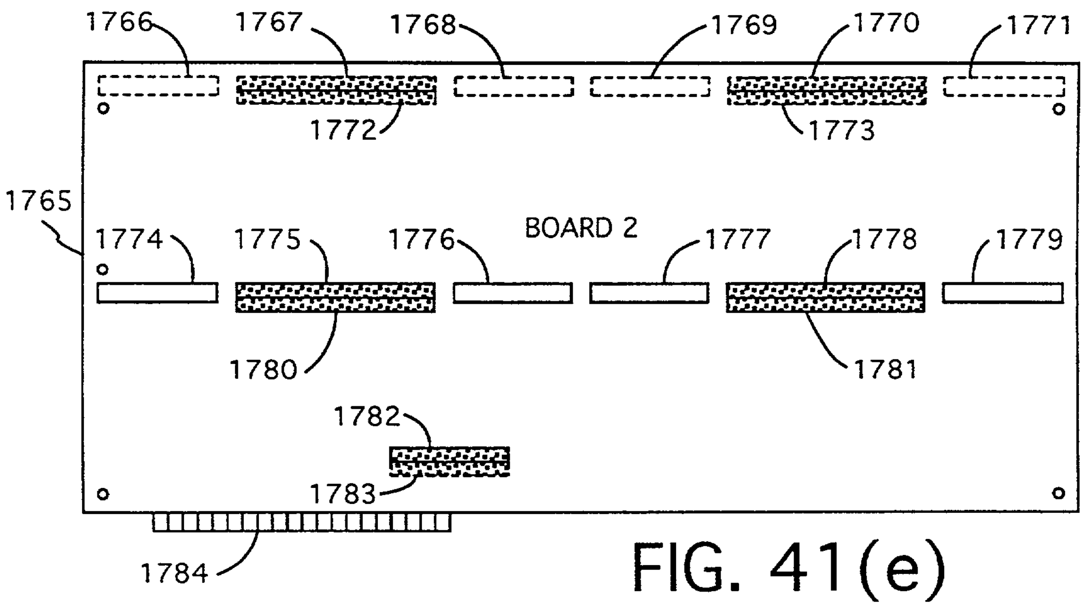


FIG. 41(e)

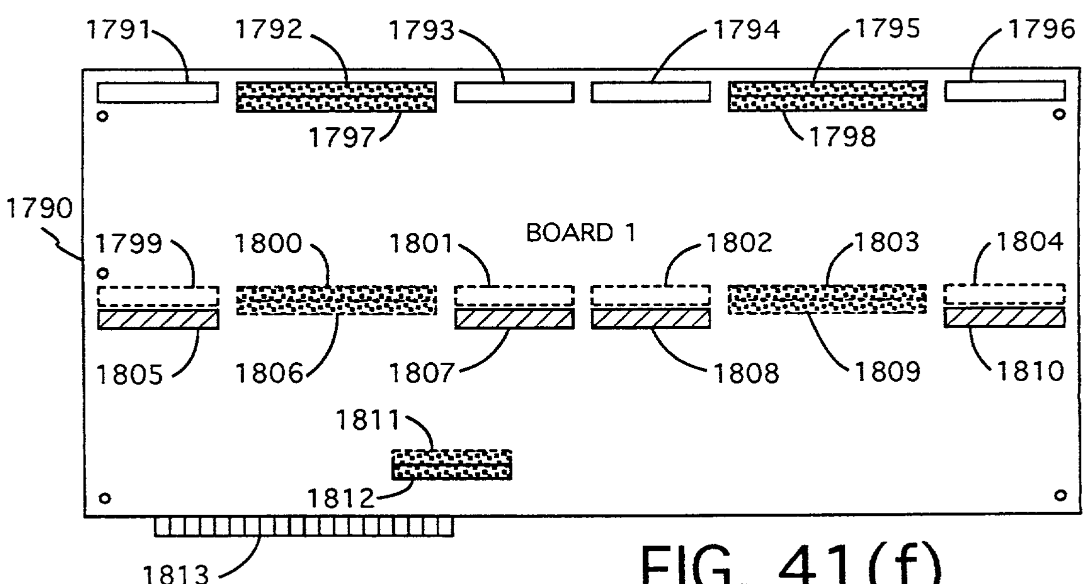


FIG. 41(f)

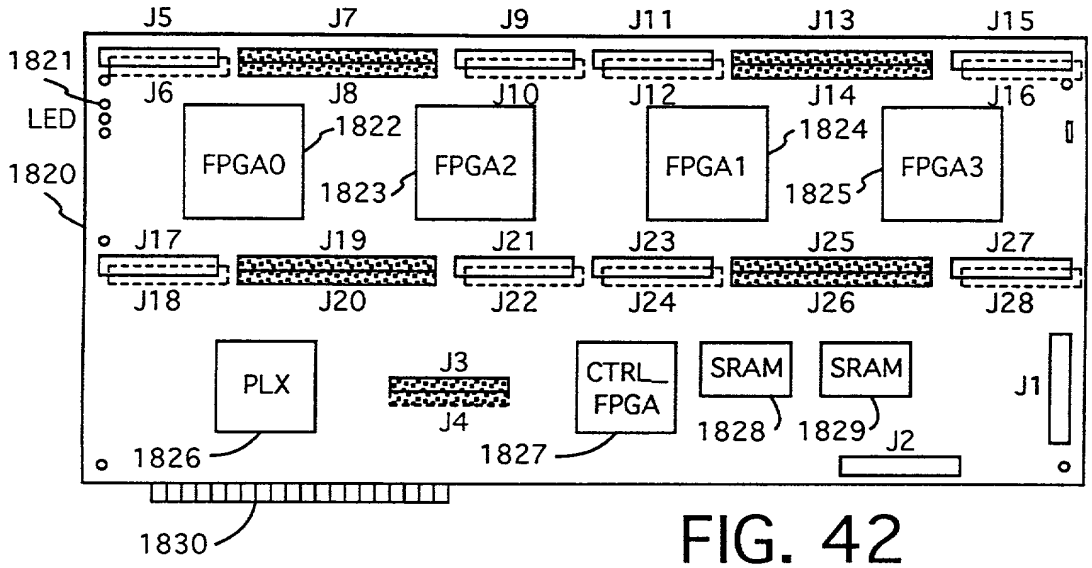


FIG. 42

1840 — 2x30 HEADER, SMD, COMPONENT SIDE

1841 — 2x30 RECEPTACLE, SMD, SOLDER SIDE

1842 — 2x45, 2x30 HEADER, THRU HOLE, COMPONENT SIDE

1843 — 2x45, 2x30 RECEPTACLE, THRU HOLE, SOLDER SIDE

1844 — R-PACK, SMD, COMPONENT SIDE

1845 — R-PACK, SMD, SOLDER SIDE

FIG. 43

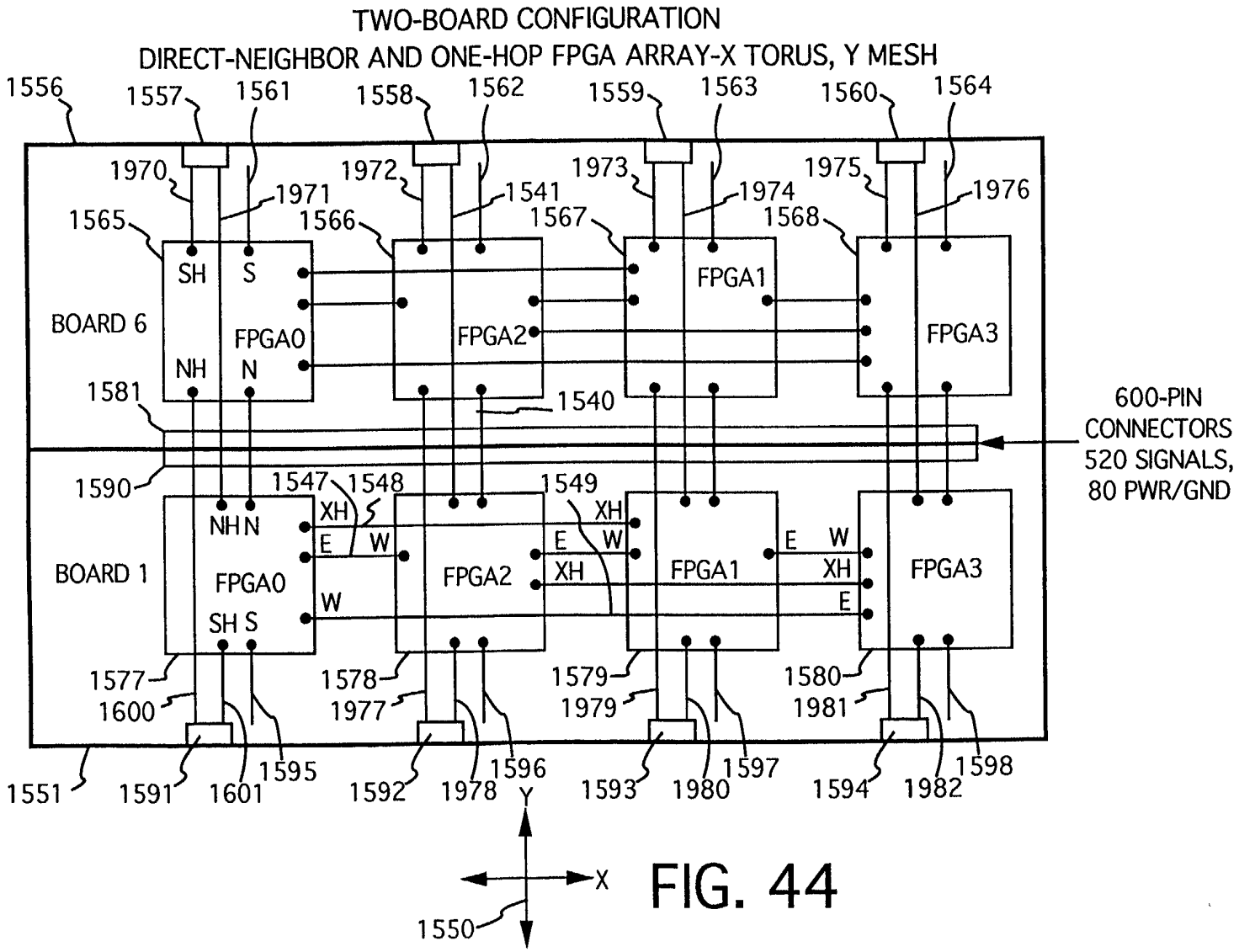

TWO-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY-X TORUS, Y MESH
1556
1557
1561
1558
1562
1559
1563
1560
1564
1970
1971
1972
1541
1973
1974
1975
1976
1565
1566
1567
1568
SH
S
FPGA0
FPGA2
FPGA1
FPGA3
BOARD 6
NH
N
1581
1540
600-PIN
CONNECTORS
520 SIGNALS,
80 PWR/GND
1590
1547
1548
1549
XH
E
W
NH N
BOARD 1
FPGA0
FPGA2
FPGA1
FPGA3
SH S
1577
1578
1579
1580
1600
1977
1979
1981
1551
1591
1601
1595
1592
1978
1596
1593
1980
1597
1594
1982
1598
Y
X
1550

FIG. 44

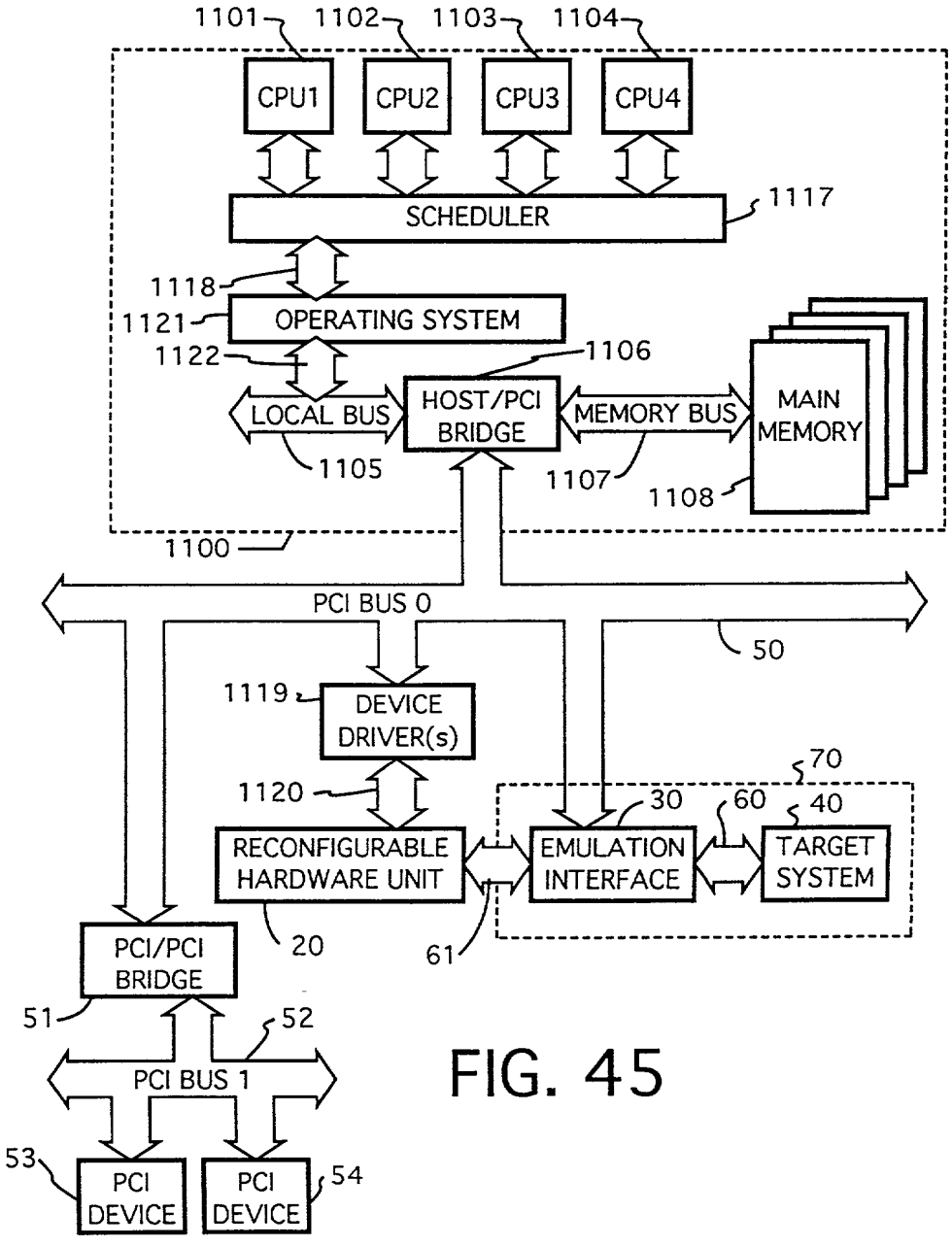

1101
1102
1103
1104
CPU1
CPU2
CPU3
CPU4
SCHEDULER
1117
1118
1121
OPERATING SYSTEM
1122
1106
LOCAL BUS
HOST/PCI BRIDGE
MEMORY BUS
MAIN MEMORY
1105
1107
1108
1100
PCI BUS 0
50
1119
DEVICE DRIVER(s)
70
1120
30
60
40
RECONFIGURABLE HARDWARE UNIT
EMULATION INTERFACE
TARGET SYSTEM
20
61
PCI/PCI BRIDGE
51
52
PCI BUS 1
53
PCI DEVICE
PCI DEVICE
54

FIG. 45

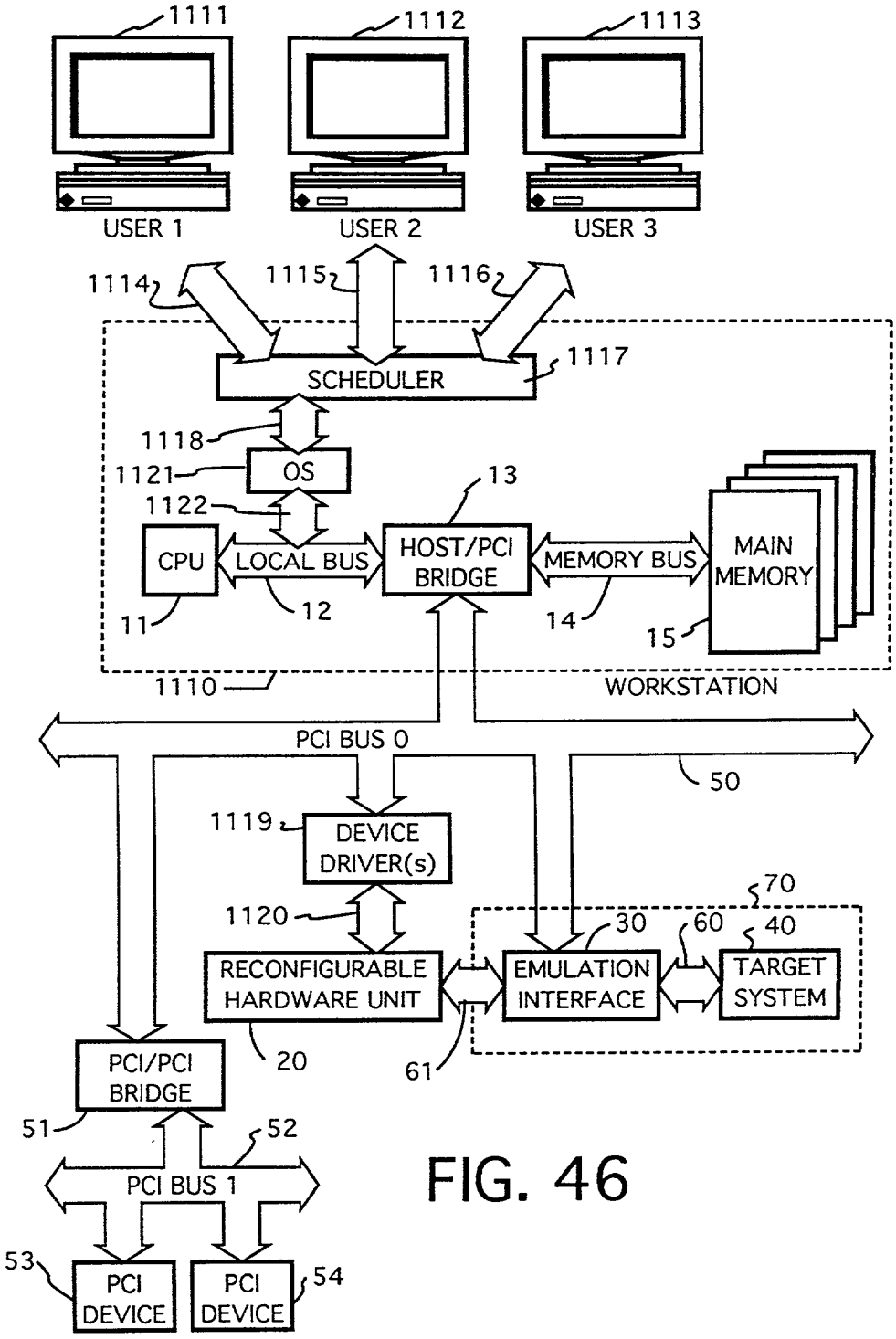

1111
1112
1113
USER 1
USER 2
USER 3
1114
1115
1116
SCHEDULER
1117
1118
OS
1121
1122
13
CPU
LOCAL BUS
HOST/PCI BRIDGE
MEMORY BUS
MAIN MEMORY
11
12
14
15
1110
WORKSTATION
PCI BUS 0
50
1119
DEVICE DRIVER(s)
70
1120
30
60
40
RECONFIGURABLE HARDWARE UNIT
EMULATION INTERFACE
TARGET SYSTEM
20
61
PCI/PCI BRIDGE
51
52
PCI BUS 1
53
PCI DEVICE
PCI DEVICE
54

FIG. 46

FOOTOT "h2T00660

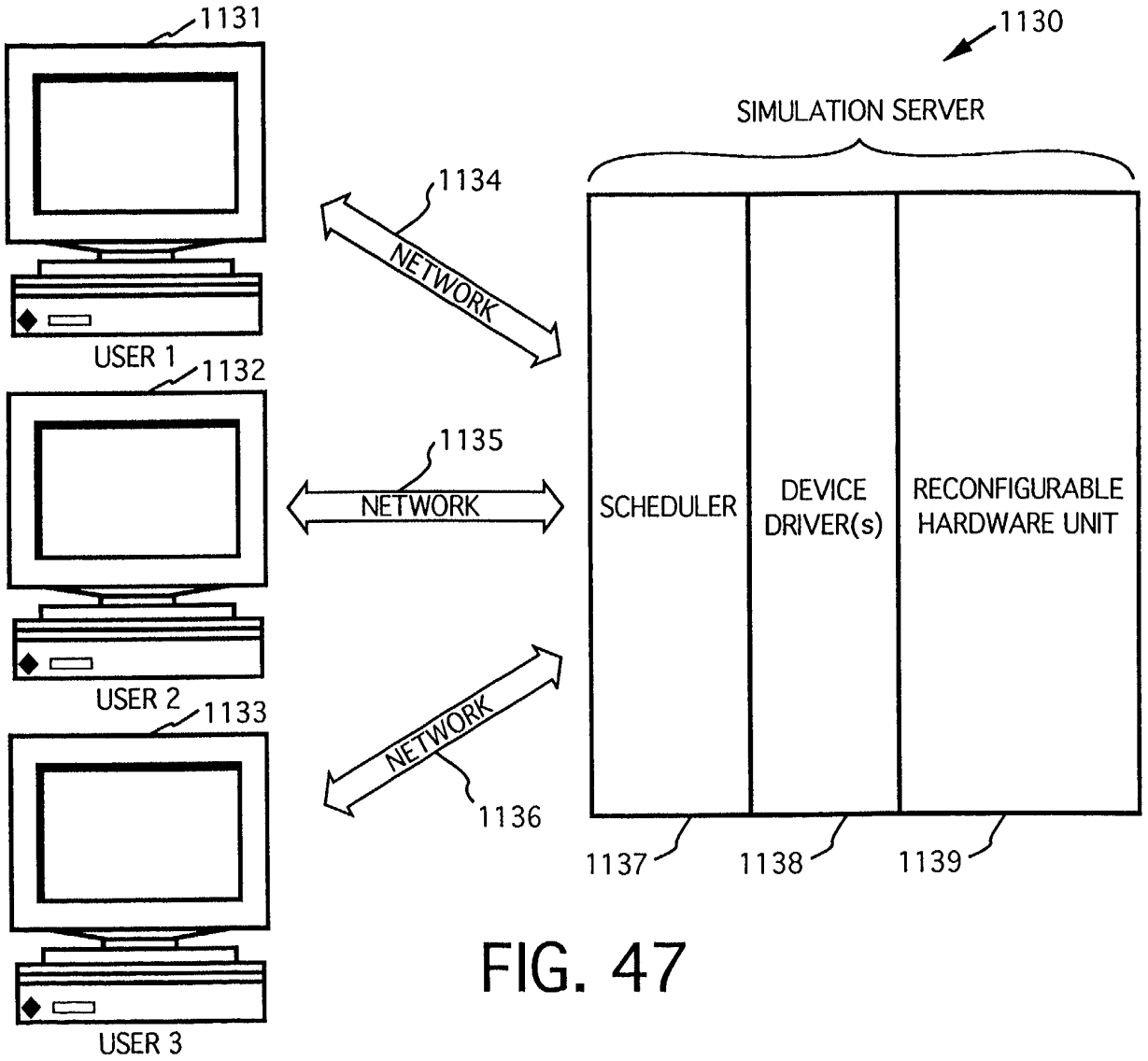


FIG. 47

FOOTOT"h2T00660

SIMULATION SERVER ARCHITECTURE

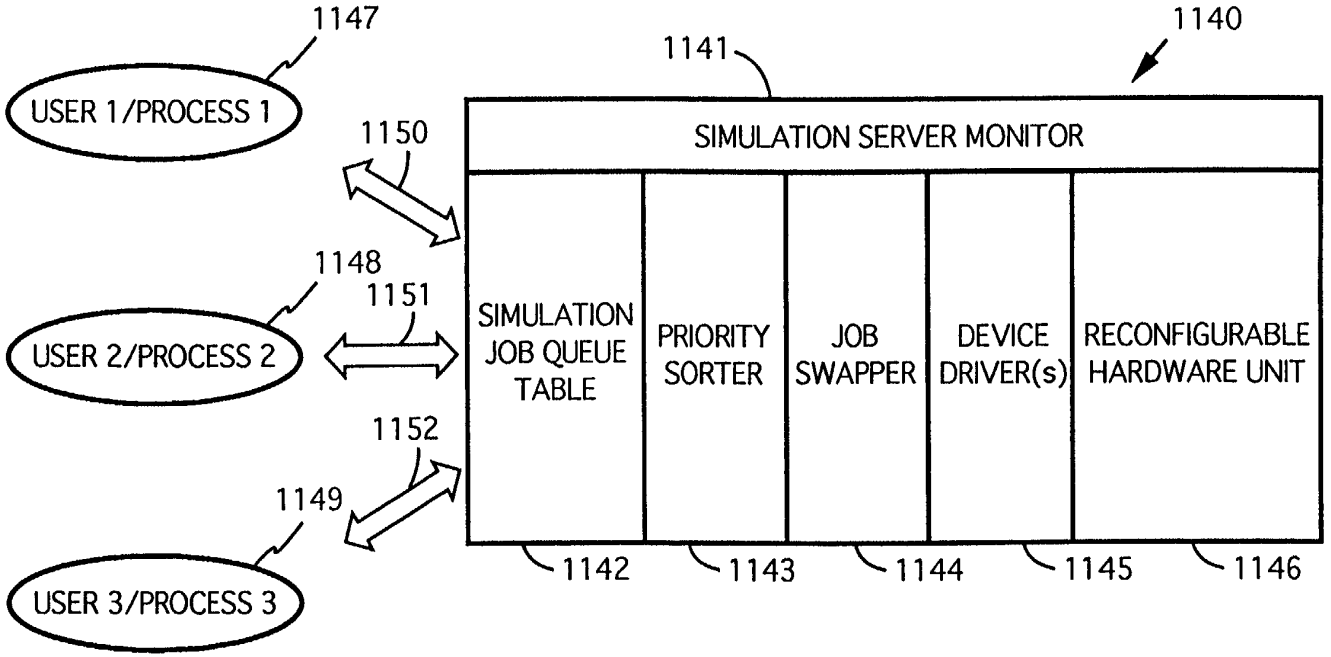


FIG. 48

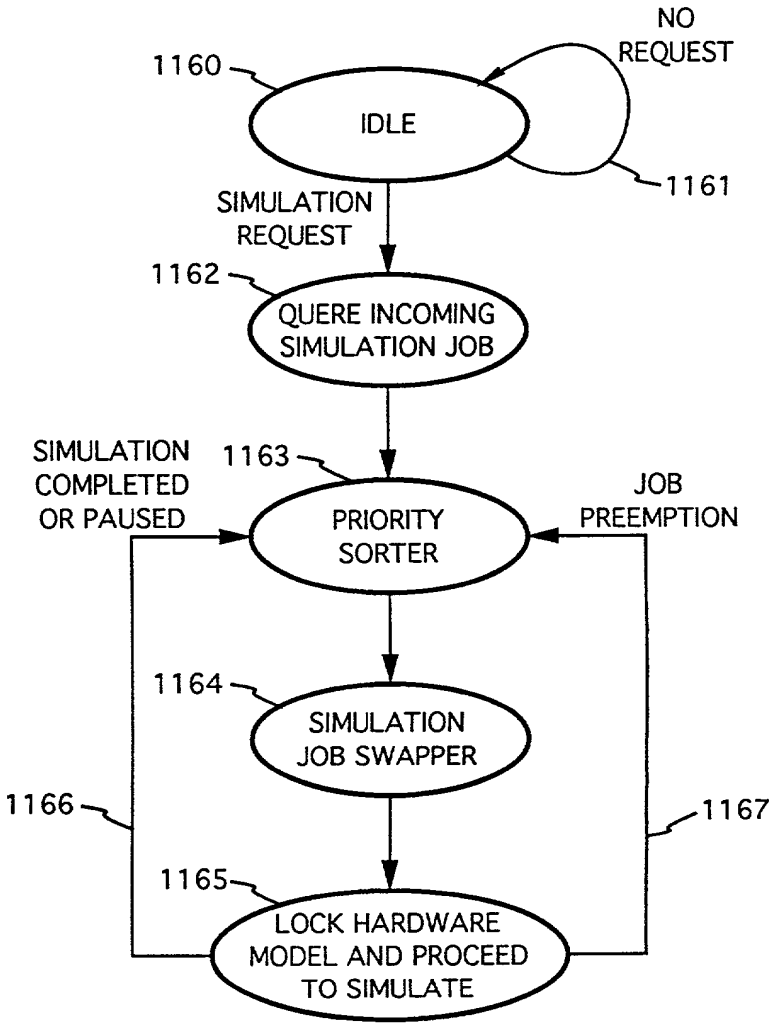

NO
REQUEST
1160
IDLE
1161
SIMULATION
REQUEST
1162
QUERE INCOMING
SIMULATION JOB
SIMULATION
COMPLETED
OR PAUSED
1163
PRIORITY
SORTER
JOB
PREEMPTION
1164
SIMULATION
JOB SWAPPER
1166
1167
1165
LOCK HARDWARE
MODEL AND PROCEED
TO SIMULATE

FIG. 49

JOB SWAPPER

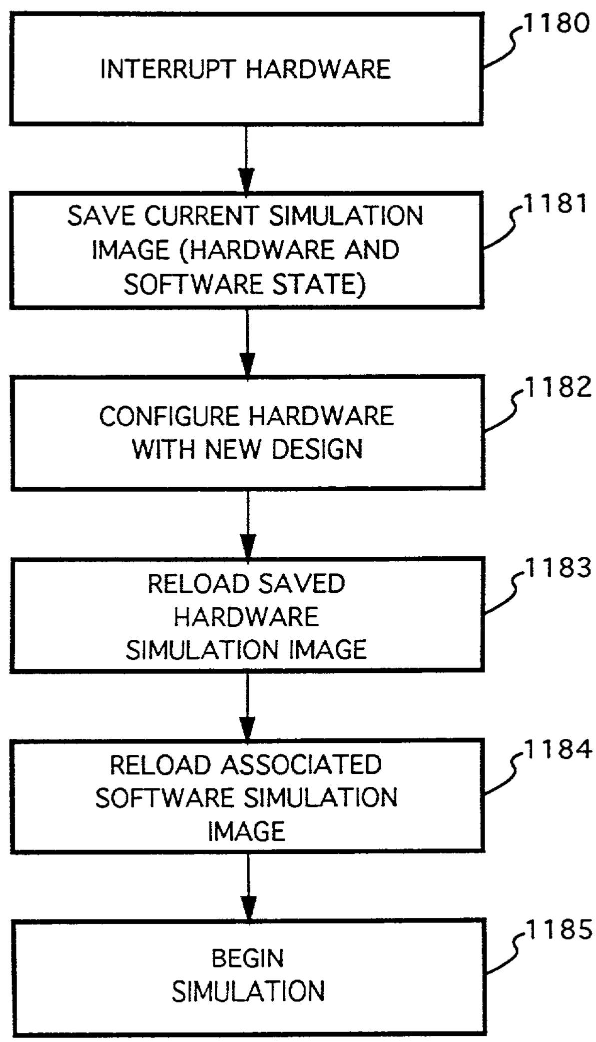


FIG. 50

Electronic Patent Solutions

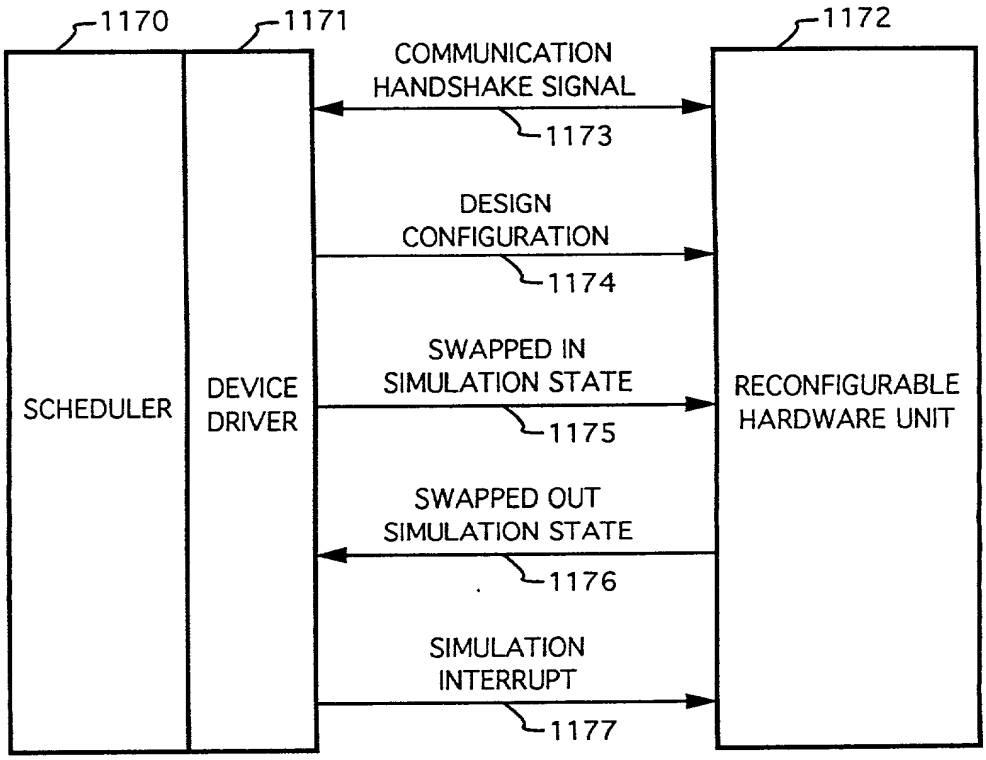


FIG. 51

PRIORITY I { JOB A, JOB B

PRIORITY II { JOB C, JOB D

TIME-SHARED HARDWARE USAGE:

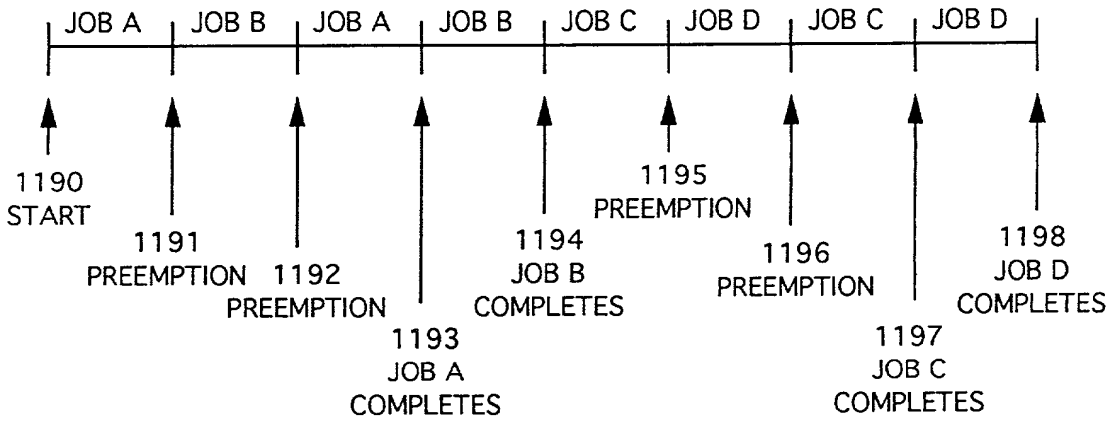


FIG. 52

Page 1 of 1 Proposed

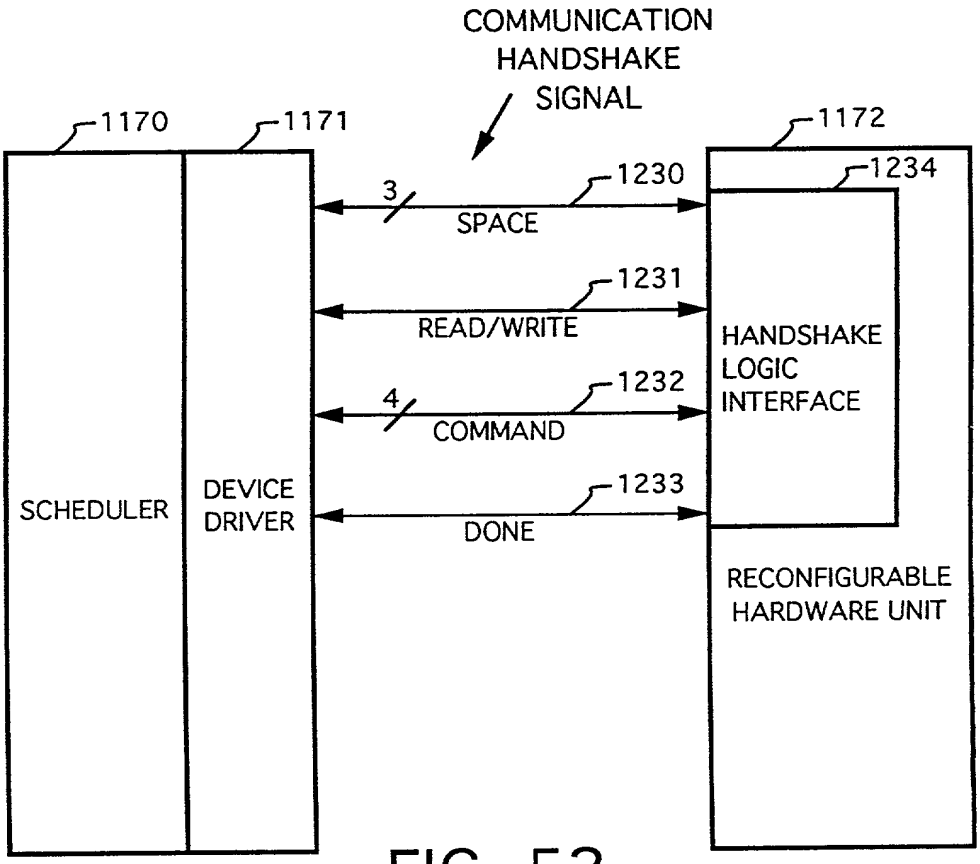


FIG. 53

COMMUNICATION
HANDSHAKE PROTOCOL

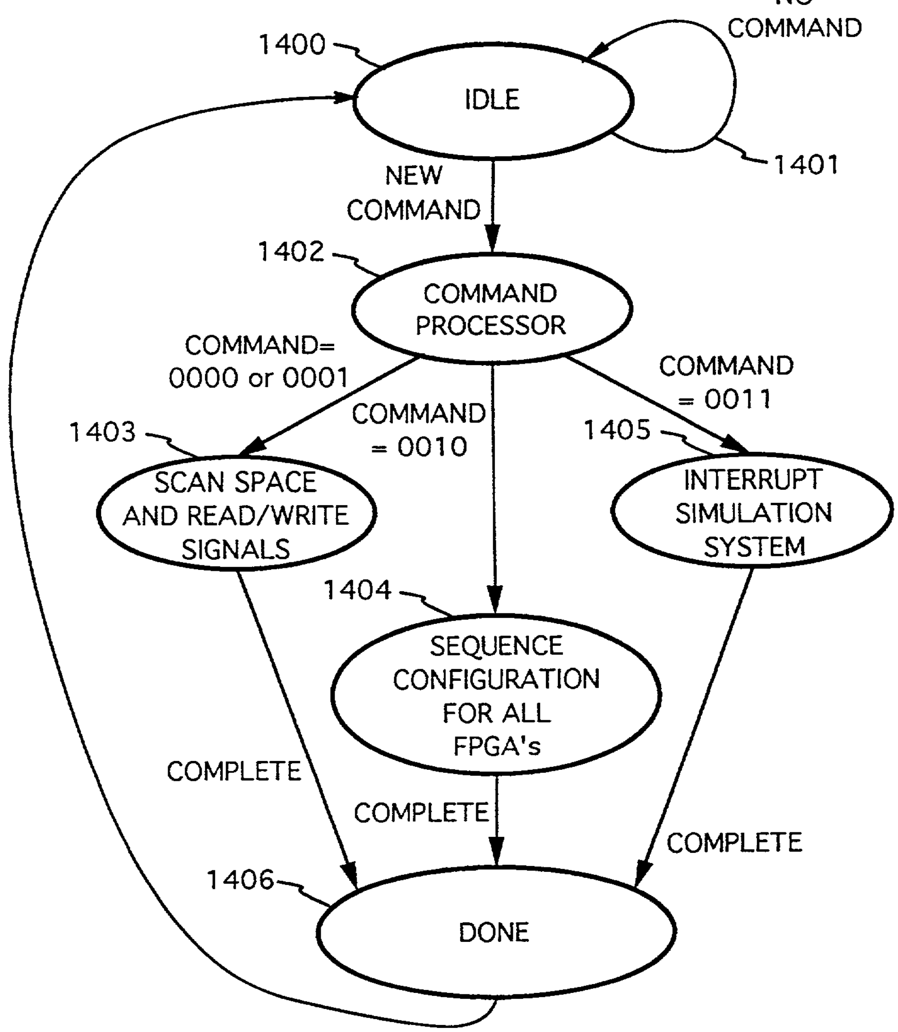


FIG. 54

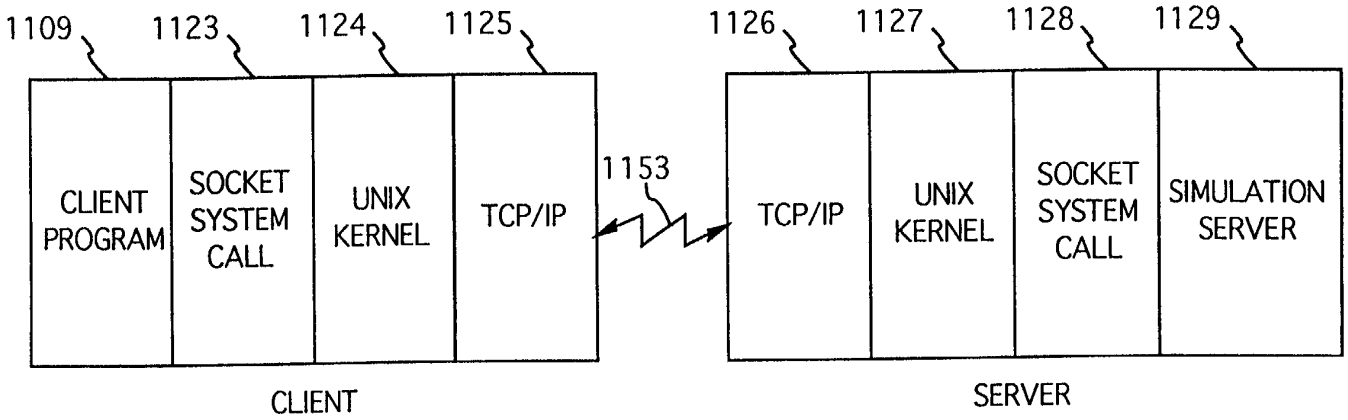

1109
1123
1124
1125
1126
1127
1128
1129
1153
CLIENT PROGRAM
SOCKET SYSTEM CALL
UNIX KERNEL
TCP/IP
TCP/IP
UNIX KERNEL
SOCKET SYSTEM CALL
SIMULATION SERVER
CLIENT
SERVER

FIG. 55

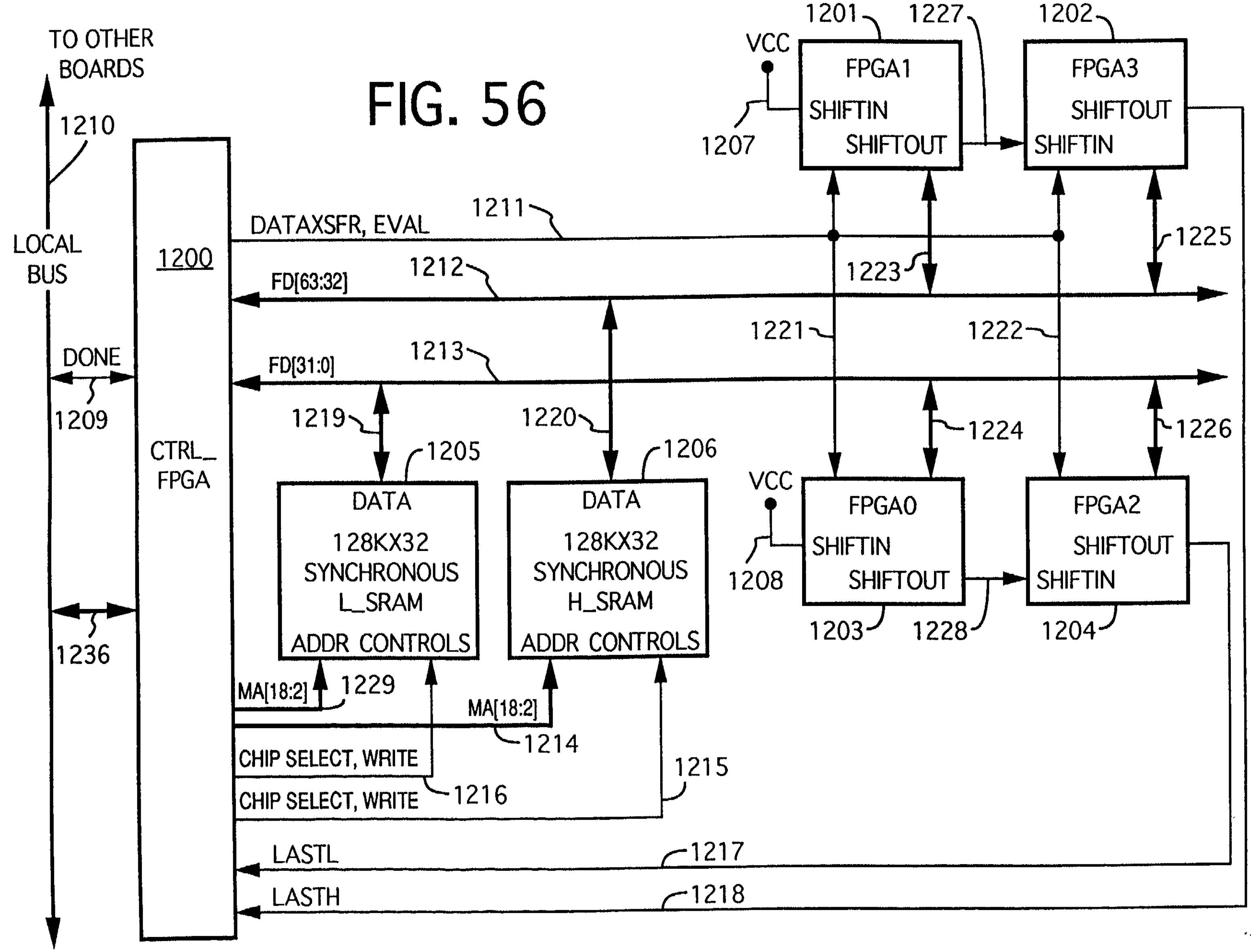

TO OTHER BOARDS
1210
LOCAL BUS
DONE
1209
1236
1200
CTRL_ FPGA
DATAXSFR, EVAL
1211
FD[63:32]
1212
FD[31:0]
1213
1219
1220
1205
1206
DATA
128KX32
SYNCHRONOUS
L_SRAM
ADDR CONTROLS
DATA
128KX32
SYNCHRONOUS
H_SRAM
ADDR CONTROLS
MA[18:2]
1229
MA[18:2]
1214
CHIP SELECT, WRITE
CHIP SELECT, WRITE
1216
1215
LASTL
1217
LASTH
1218
VCC
1207
1201
FPGA1
SHIFTIN
SHIFTOUT
1227
1202
FPGA3
SHIFTOUT
SHIFTIN
1223
1225
1221
1222
1224
1226
VCC
1208
FPGA0
SHIFTIN
SHIFTOUT
1203
1228
FPGA2
SHIFTOUT
SHIFTIN
1204

FIG. 56

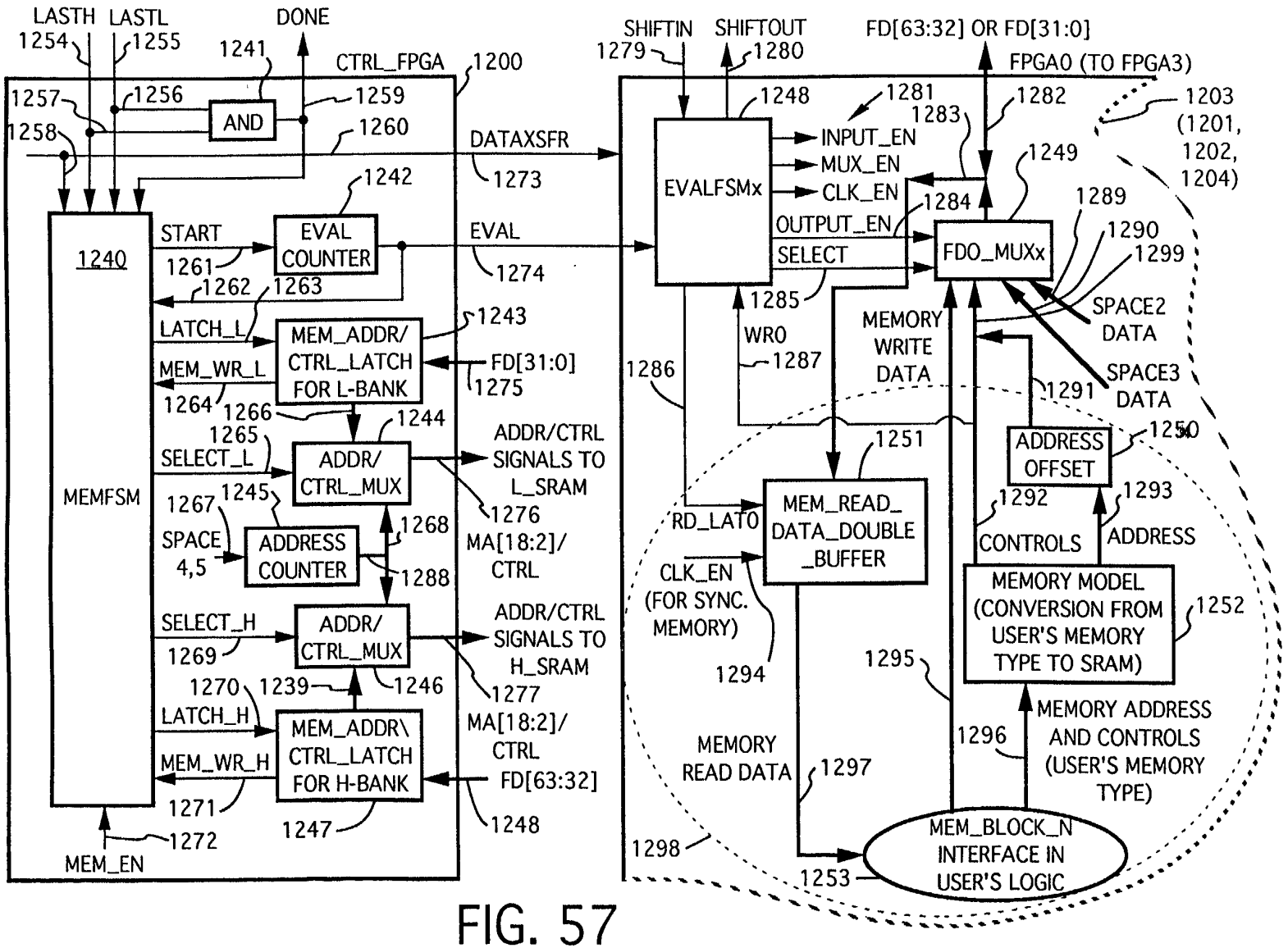

LASTH
1254
LASTL
1255
DONE
1241
CTRL_FPGA
1200
1257
1256
AND
1259
1258
1260
DATAXSFR
1273
1242
1240
START
1261
EVAL
COUNTER
EVAL
1274
1262
1263
LATCH_L
MEM_ADDR/
CTRL_LATCH
FOR L-BANK
1243
MEM_WR_L
1264
FD[31:0]
1275
1266
1244
1265
SELECT_L
ADDR/
CTRL_MUX
ADDR/CTRL
SIGNALS TO
L_SRAM
1276
MEMFSM
1267
1245
SPACE
4,5
ADDRESS
COUNTER
1268
MA[18:2]/
CTRL
1288
SELECT_H
1269
ADDR/
CTRL_MUX
ADDR/CTRL
SIGNALS TO
H_SRAM
1277
1270
1239
1246
LATCH_H
MEM_ADDR\
CTRL_LATCH
FOR H-BANK
MA[18:2]/
CTRL
MEM_WR_H
1271
FD[63:32]
1248
1247
1272
MEM_EN
SHIFTIN
1279
SHIFTOUT
1280
FD[63:32] OR FD[31:0]
FPGA0 (TO FPGA3)
1248
1281
1283
1282
1203
(1201,
1202,
1204)
EVALFSMx
INPUT_EN
MUX_EN
CLK_EN
OUTPUT_EN
SELECT
1284
1249
1289
1290
1299
FDO_MUXx
1285
SPACE2
DATA
WR0
1287
MEMORY
WRITE
DATA
SPACE3
DATA
1286
1291
1250
ADDRESS
OFFSET
1251
MEM_READ_
DATA_DOUBLE
_BUFFER
1292
1293
RD_LAT0
CONTROLS
ADDRESS
CLK_EN
(FOR SYNC.
MEMORY)
MEMORY MODEL
(CONVERSION FROM
USER'S MEMORY
TYPE TO SRAM)
1252
1294
1295
MEMORY ADDRESS
AND CONTROLS
(USER'S MEMORY
TYPE)
1296
MEMORY
READ DATA
1297
1298
MEM_BLOCK_N
INTERFACE IN
USER'S LOGIC
1253

FIG. 57

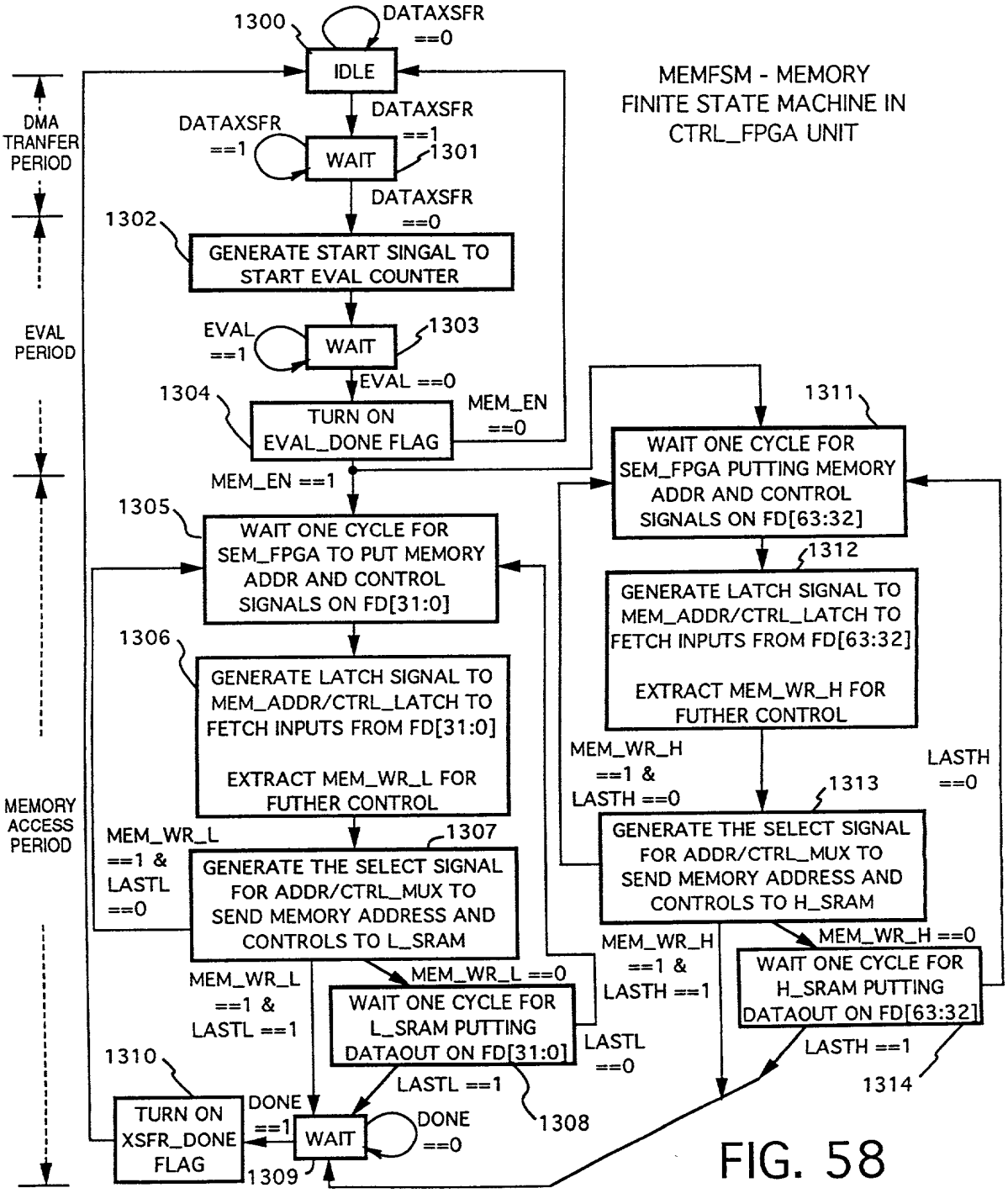

MEMFSM - MEMORY FINITE STATE MACHINE IN CTRL_FPGA UNIT
DMA TRANFER PERIOD
EVAL PERIOD
MEMORY ACCESS PERIOD
1300
DATAXSFR ==0
IDLE
DATAXSFR ==1
DATAXSFR ==1
WAIT
1301
DATAXSFR ==0
1302
GENERATE START SINGAL TO START EVAL COUNTER
EVAL ==1
WAIT
1303
EVAL ==0
1304
TURN ON EVAL_DONE FLAG
MEM_EN ==0
MEM_EN ==1
1305
WAIT ONE CYCLE FOR SEM_FPGA TO PUT MEMORY ADDR AND CONTROL SIGNALS ON FD[31:0]
1306
GENERATE LATCH SIGNAL TO MEM_ADDR/CTRL_LATCH TO FETCH INPUTS FROM FD[31:0]
EXTRACT MEM_WR_L FOR FUTHER CONTROL
MEM_WR_L ==1 & LASTL ==0
1307
GENERATE THE SELECT SIGNAL FOR ADDR/CTRL_MUX TO SEND MEMORY ADDRESS AND CONTROLS TO L_SRAM
MEM_WR_L ==1 & LASTL ==1
MEM_WR_L ==0
WAIT ONE CYCLE FOR L_SRAM PUTTING DATAOUT ON FD[31:0]
LASTL ==0
LASTL ==1
1308
1310
TURN ON XSFR_DONE FLAG
DONE ==1
WAIT
DONE ==0
1309
1311
WAIT ONE CYCLE FOR SEM_FPGA PUTTING MEMORY ADDR AND CONTROL SIGNALS ON FD[63:32]
1312
GENERATE LATCH SIGNAL TO MEM_ADDR/CTRL_LATCH TO FETCH INPUTS FROM FD[63:32]
EXTRACT MEM_WR_H FOR FUTHER CONTROL
MEM_WR_H ==1 & LASTH ==0
LASTH ==0
1313
GENERATE THE SELECT SIGNAL FOR ADDR/CTRL_MUX TO SEND MEMORY ADDRESS AND CONTROLS TO H_SRAM
MEM_WR_H ==1 & LASTH ==1
MEM_WR_H ==0
WAIT ONE CYCLE FOR H_SRAM PUTTING DATAOUT ON FD[63:32]
LASTH ==1
1314

FIG. 58

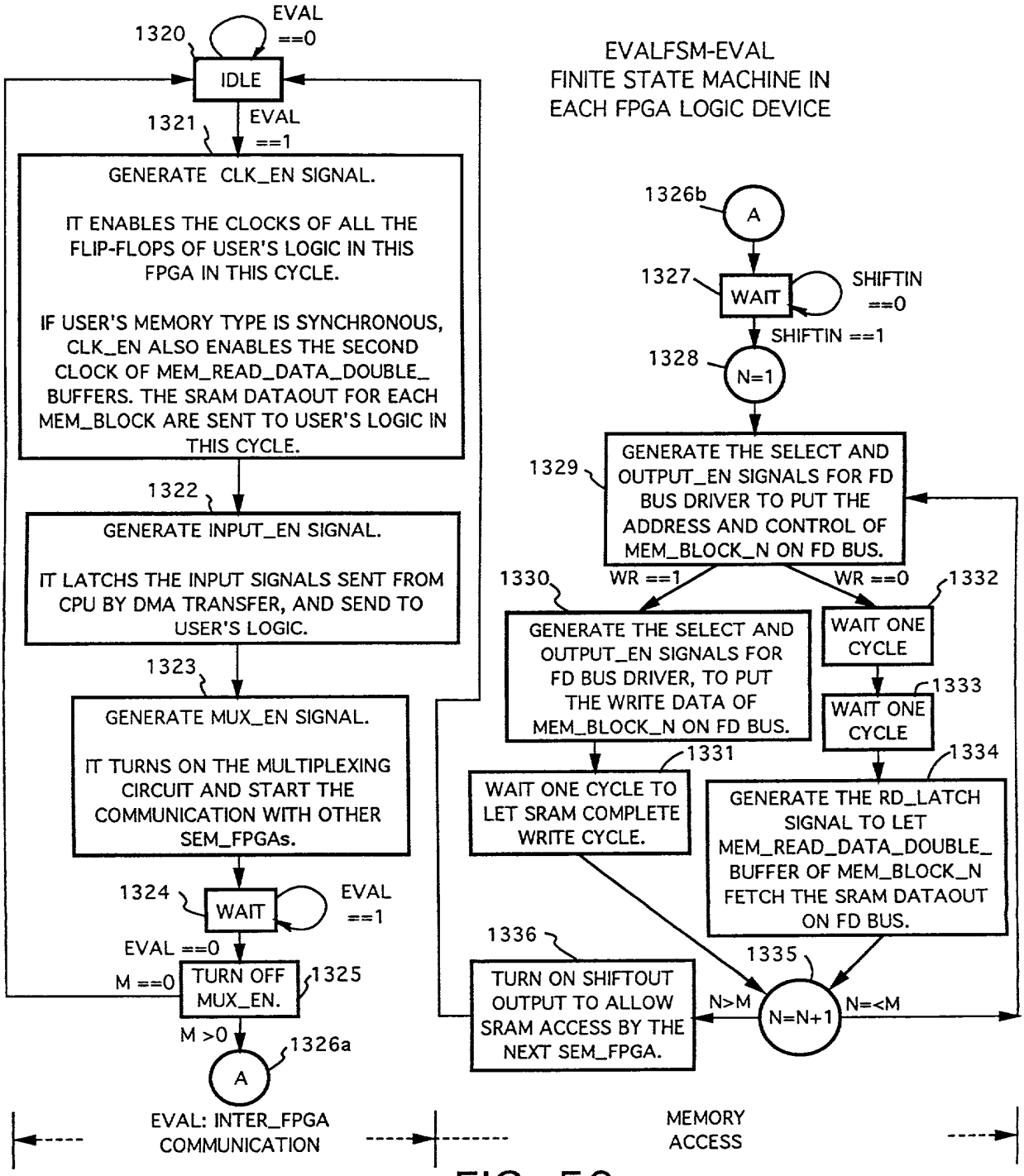

EVALFSM-EVAL
FINITE STATE MACHINE IN
EACH FPGA LOGIC DEVICE
1320
EVAL
==0
IDLE
1321
EVAL
==1
GENERATE CLK_EN SIGNAL.
IT ENABLES THE CLOCKS OF ALL THE FLIP-FLOPS OF USER'S LOGIC IN THIS FPGA IN THIS CYCLE.
IF USER'S MEMORY TYPE IS SYNCHRONOUS, CLK_EN ALSO ENABLES THE SECOND CLOCK OF MEM_READ_DATA_DOUBLE_ BUFFERS. THE SRAM DATAOUT FOR EACH MEM_BLOCK ARE SENT TO USER'S LOGIC IN THIS CYCLE.
1322
GENERATE INPUT_EN SIGNAL.
IT LATCHS THE INPUT SIGNALS SENT FROM CPU BY DMA TRANSFER, AND SEND TO USER'S LOGIC.
1323
GENERATE MUX_EN SIGNAL.
IT TURNS ON THE MULTIPLEXING CIRCUIT AND START THE COMMUNICATION WITH OTHER SEM_FPGAs.
1324
WAIT
EVAL
==1
EVAL ==0
M ==0
TURN OFF MUX_EN.
1325
M >0
1326a
A
EVAL: INTER_FPGA COMMUNICATION
1326b
A
1327
WAIT
SHIFTIN
==0
SHIFTIN ==1
1328
N=1
1329
GENERATE THE SELECT AND OUTPUT_EN SIGNALS FOR FD BUS DRIVER TO PUT THE ADDRESS AND CONTROL OF MEM_BLOCK_N ON FD BUS.
1330
WR ==1
WR ==0
1332
GENERATE THE SELECT AND OUTPUT_EN SIGNALS FOR FD BUS DRIVER, TO PUT THE WRITE DATA OF MEM_BLOCK_N ON FD BUS.
WAIT ONE CYCLE
1333
WAIT ONE CYCLE
1331
WAIT ONE CYCLE TO LET SRAM COMPLETE WRITE CYCLE.
1334
GENERATE THE RD_LATCH SIGNAL TO LET MEM_READ_DATA_DOUBLE_ BUFFER OF MEM_BLOCK_N FETCH THE SRAM DATAOUT ON FD BUS.
1336
TURN ON SHIFTOUT OUTPUT TO ALLOW SRAM ACCESS BY THE NEXT SEM_FPGA.
1335
N>M
N=N+1
N=<M
MEMORY ACCESS

FIG. 59

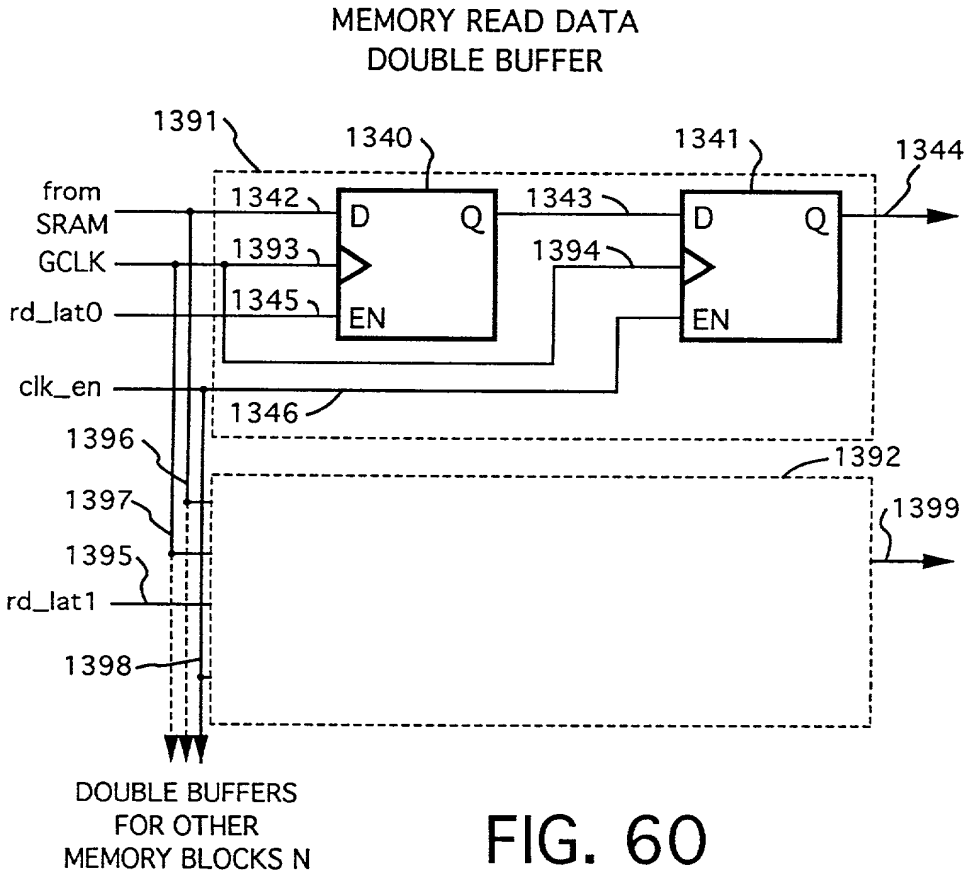

MEMORY READ DATA
DOUBLE BUFFER
1391
1340
1341
1344
from
SRAM
1342
D
Q
1343
D
Q
GCLK
1393
1394
rd_lat0
1345
EN
EN
clk_en
1346
1396
1392
1397
1399
1395
rd_lat1
1398
DOUBLE BUFFERS
FOR OTHER
MEMORY BLOCKS N

FIG. 60

TOOTOT"h2T00660

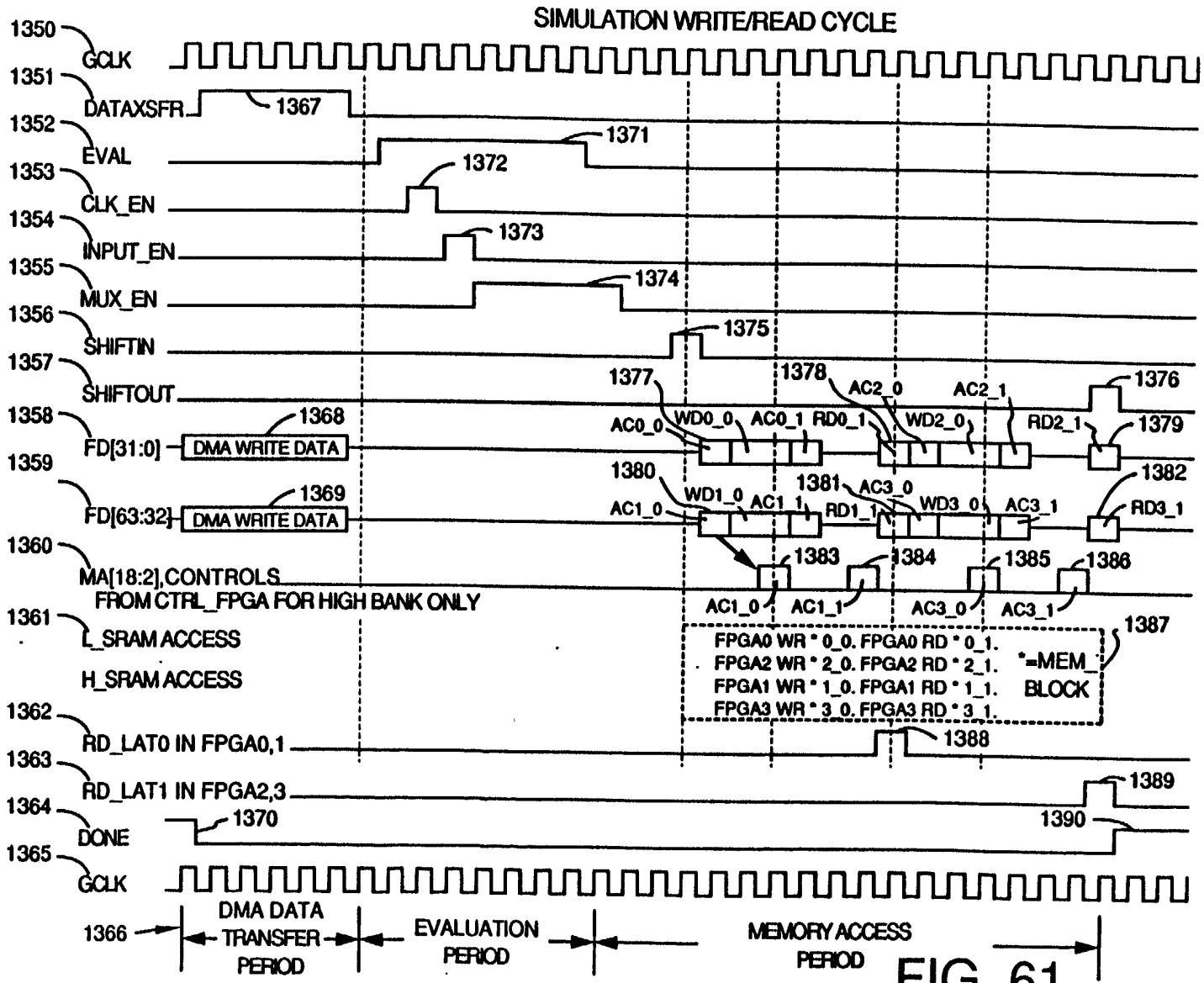


FIG. 61

TOOTOF "h2700660

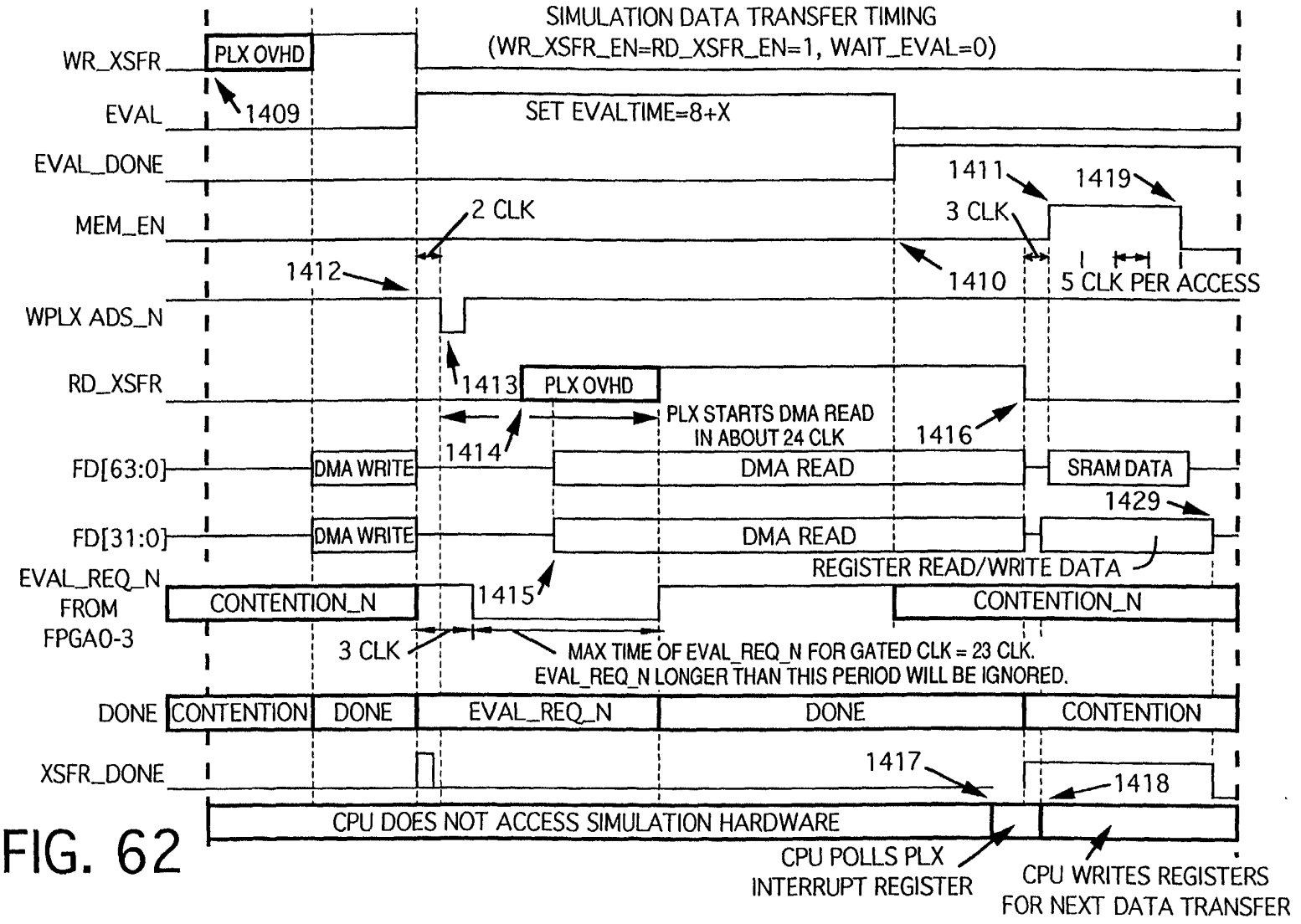


FIG. 62

T00T0T"h2T00660

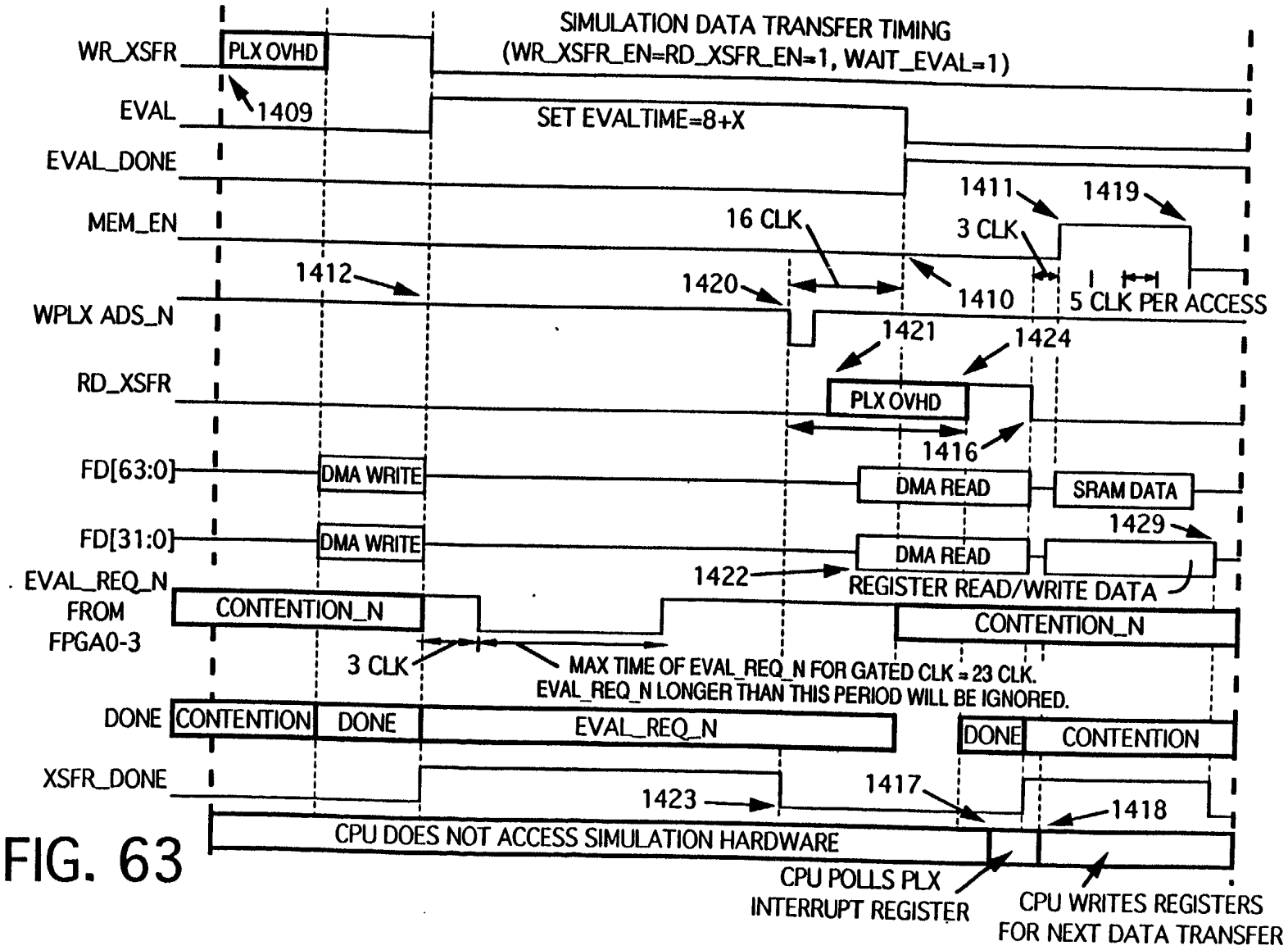


FIG. 63

TOTOT"h2T00660

Typical User Design of PCI Add-on Cards

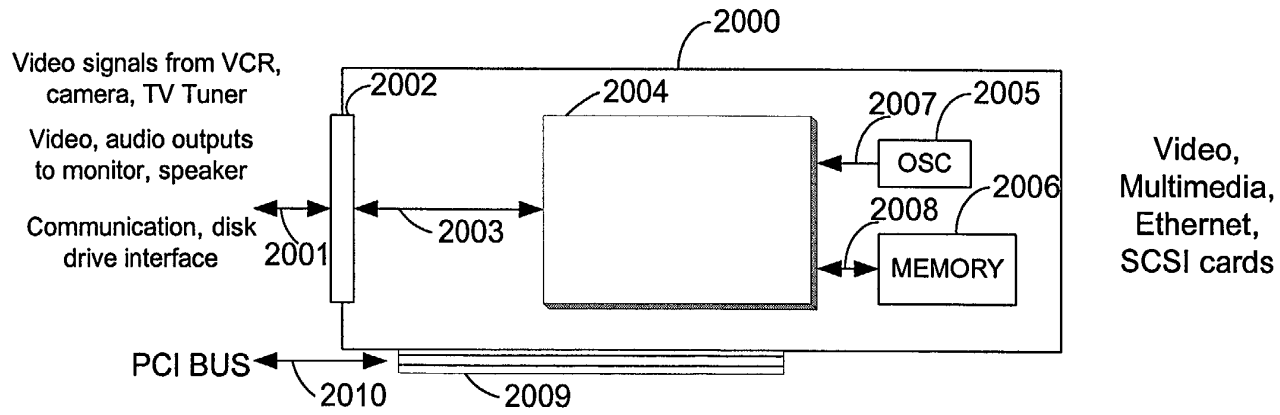


FIG. 64

Typical Hardware/Software Co-Verification

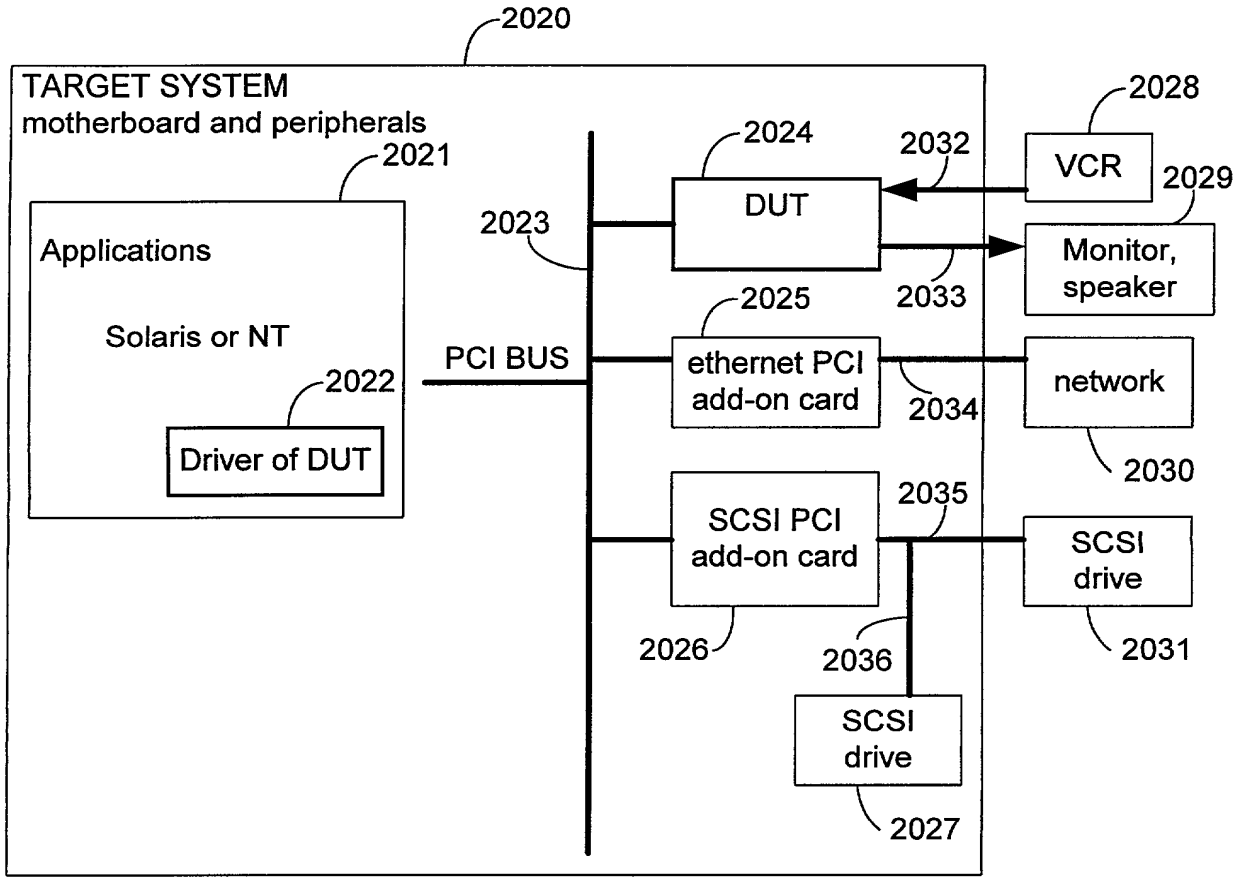
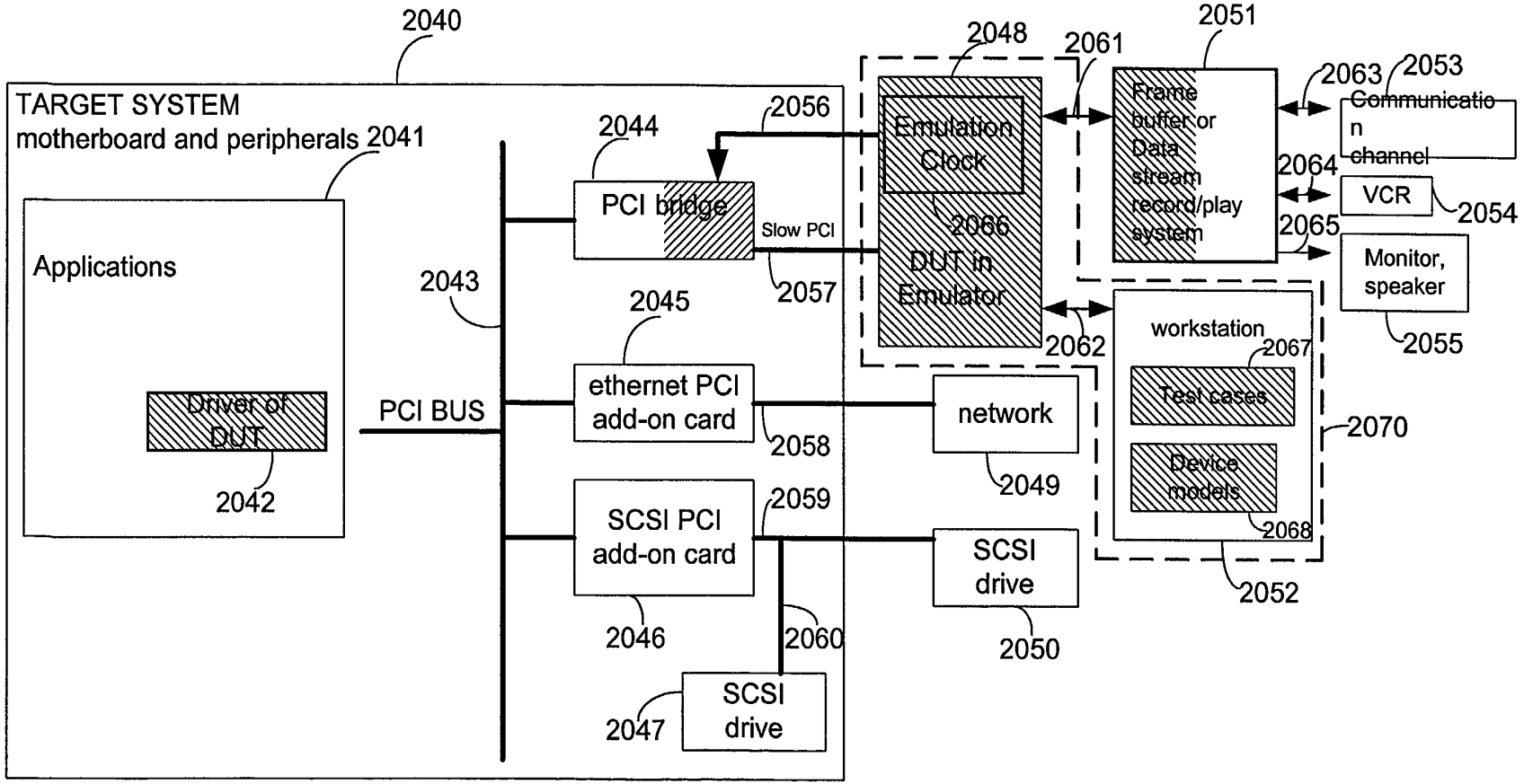


FIG. 65

Typical Co-Verification by Using Emulator



: **running time at emulation speed**

The rest of the target system is running at full speed.

FIG. 66

SIMULATION

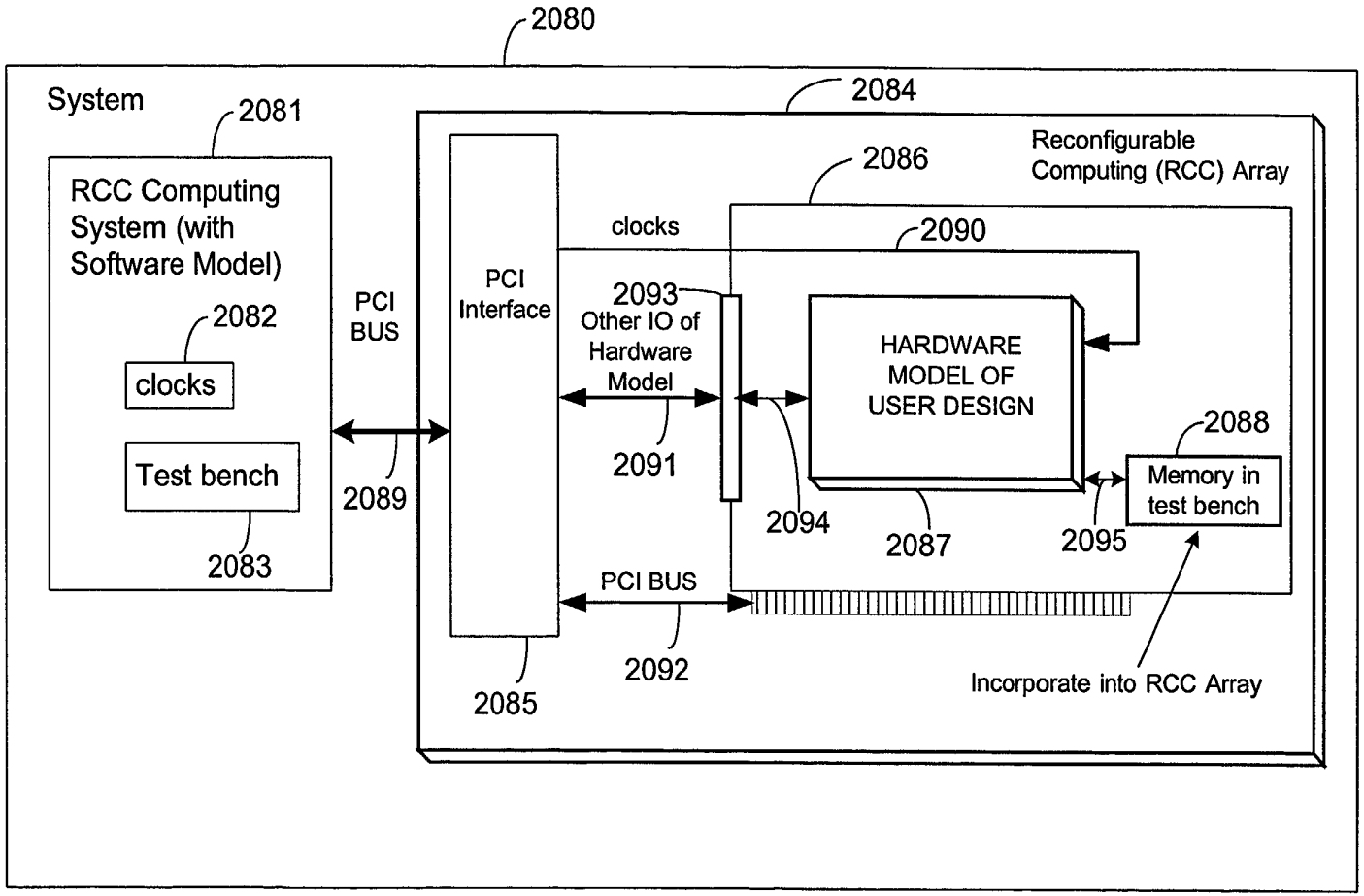


FIG. 67

CO-VERIFICATION WITHOUT EXTERNAL I/O

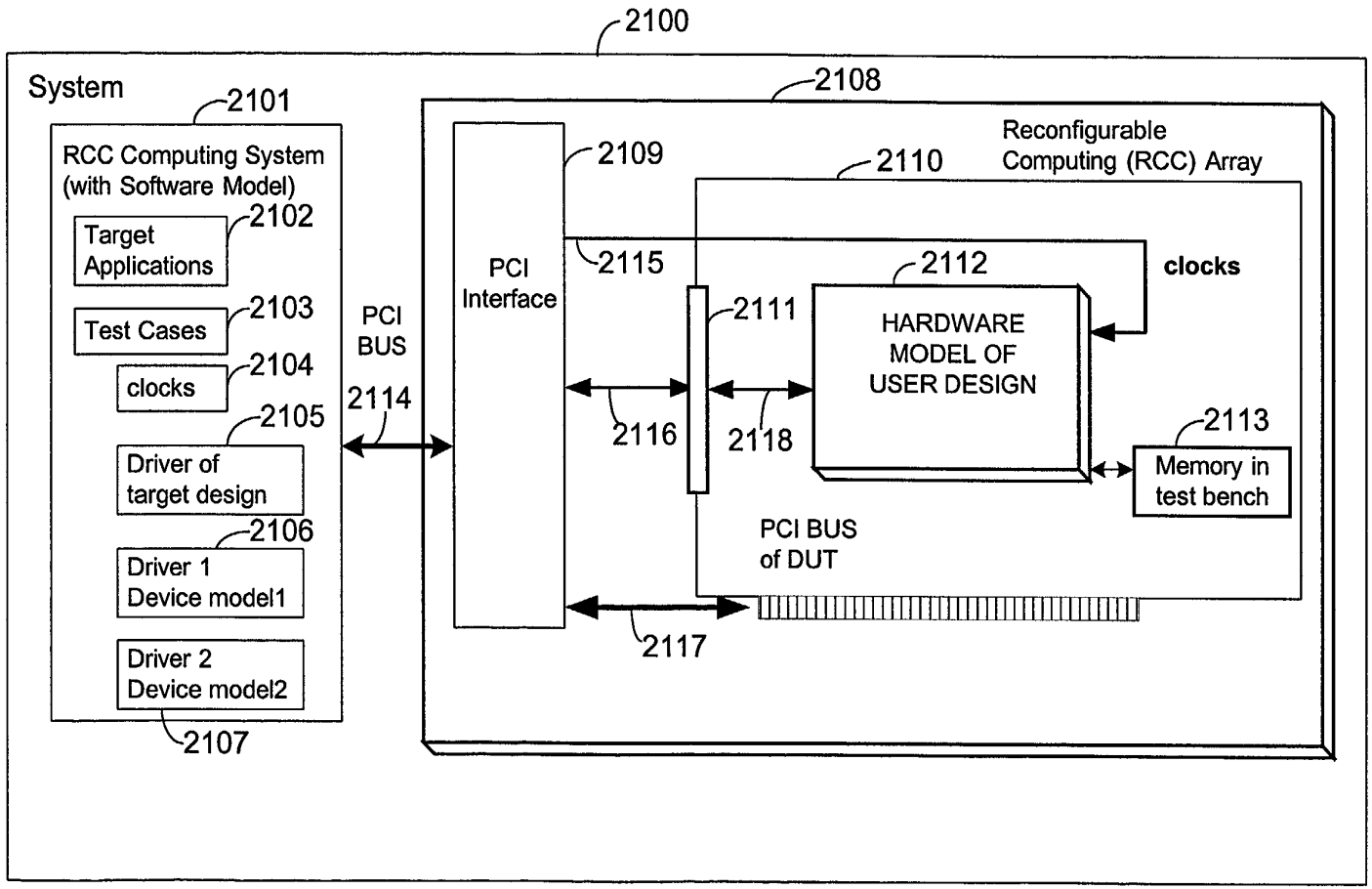


FIG. 68

CO-VERIFICATION WITH EXTERNAL I/O

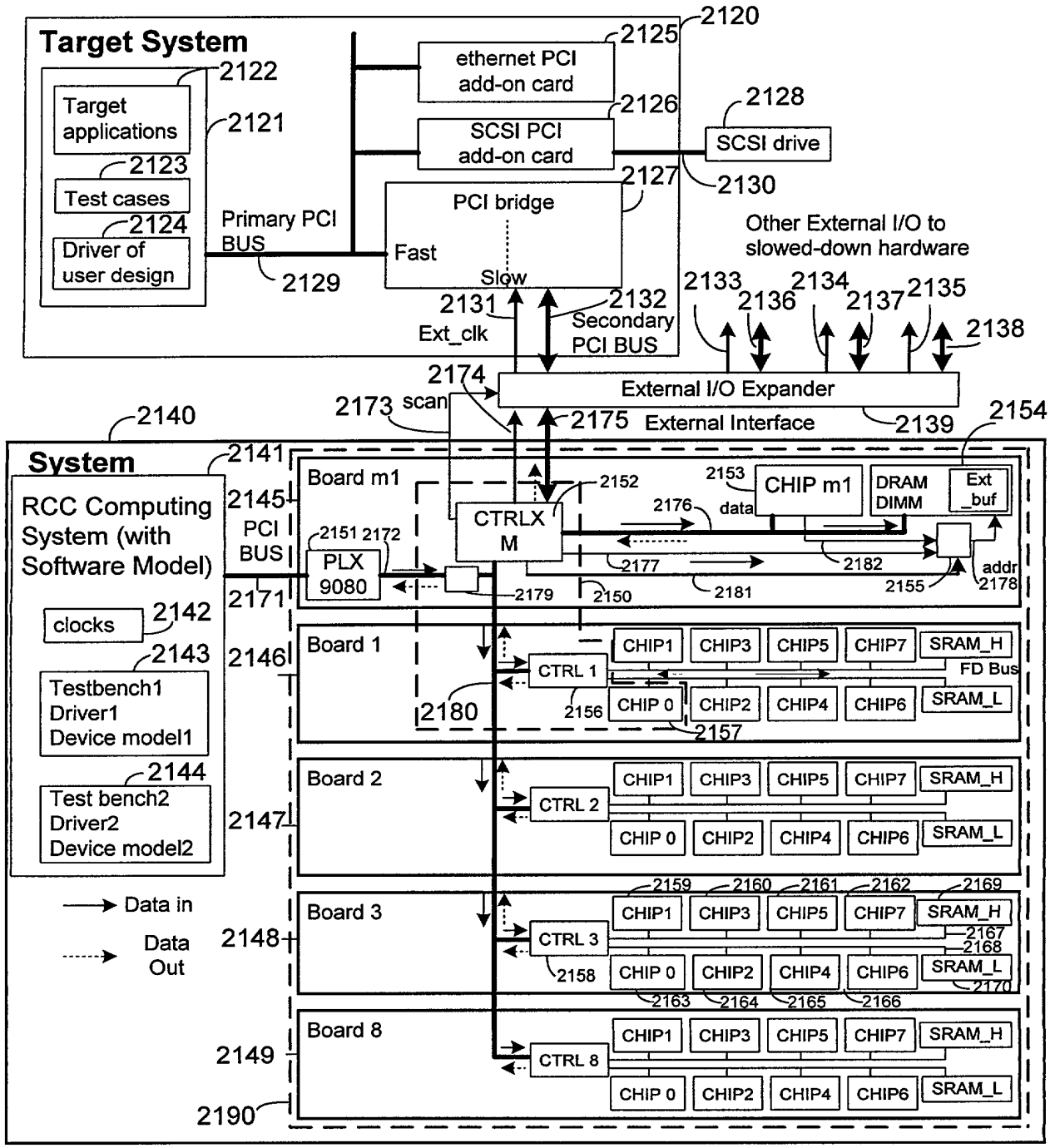


FIG. 69

CONTROL OF DATA-IN CYCLE

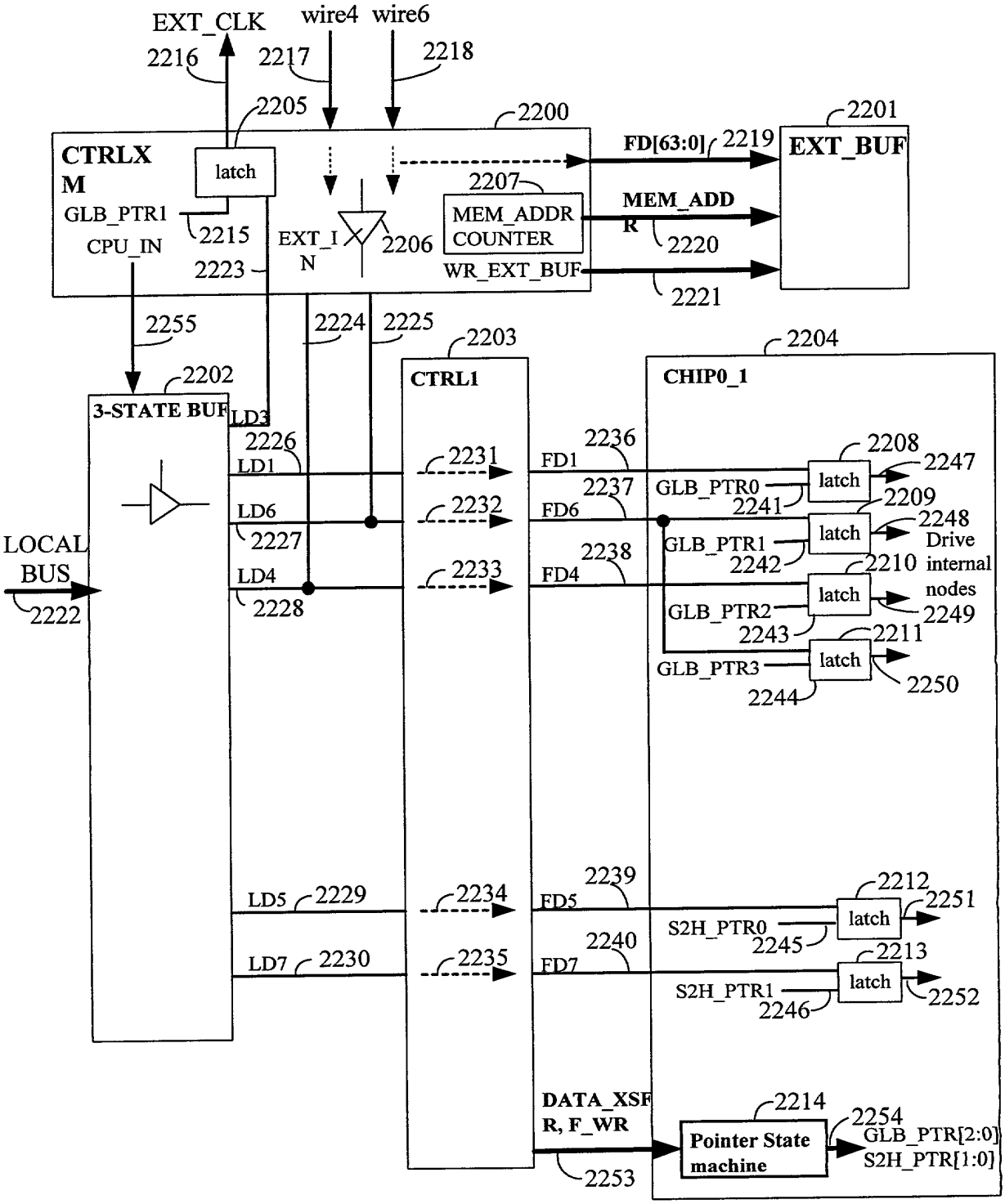


FIG. 70

CONTROL OF DATA-OUT CYCLE

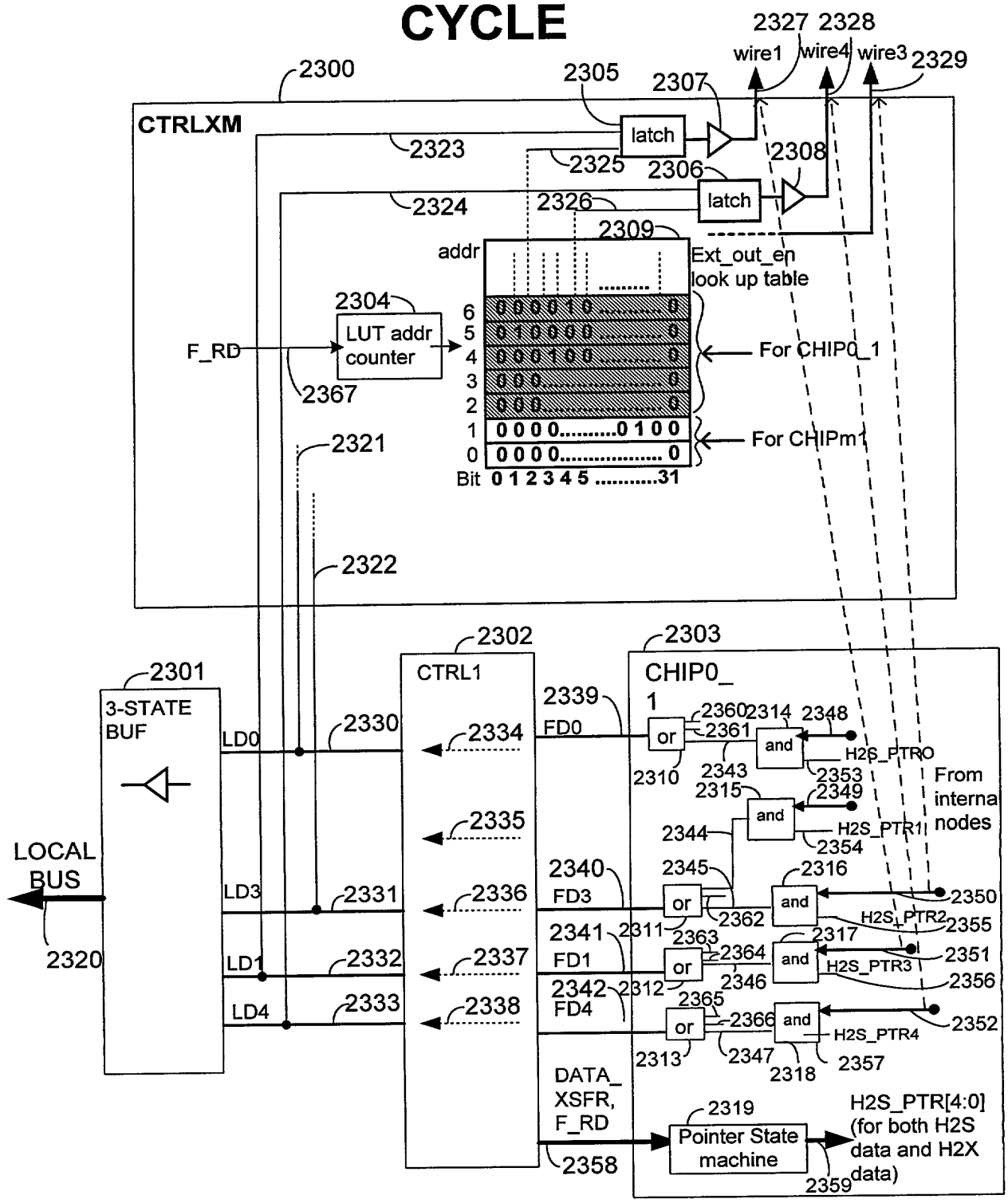


FIG. 71

CONTROL OF DATA-IN CYCLE

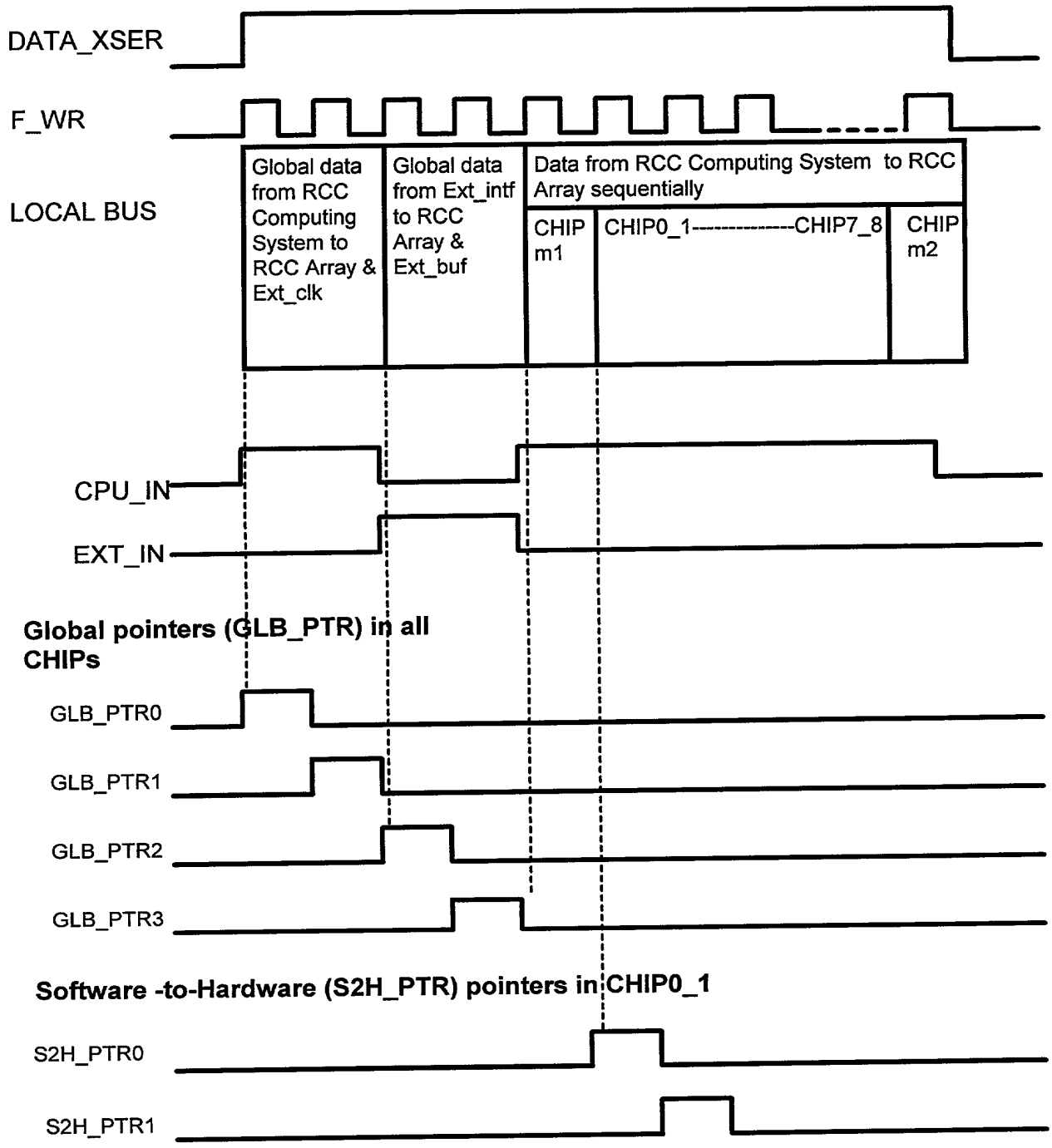


FIG. 72

CONTROL OF DATA-OUT CYCLE

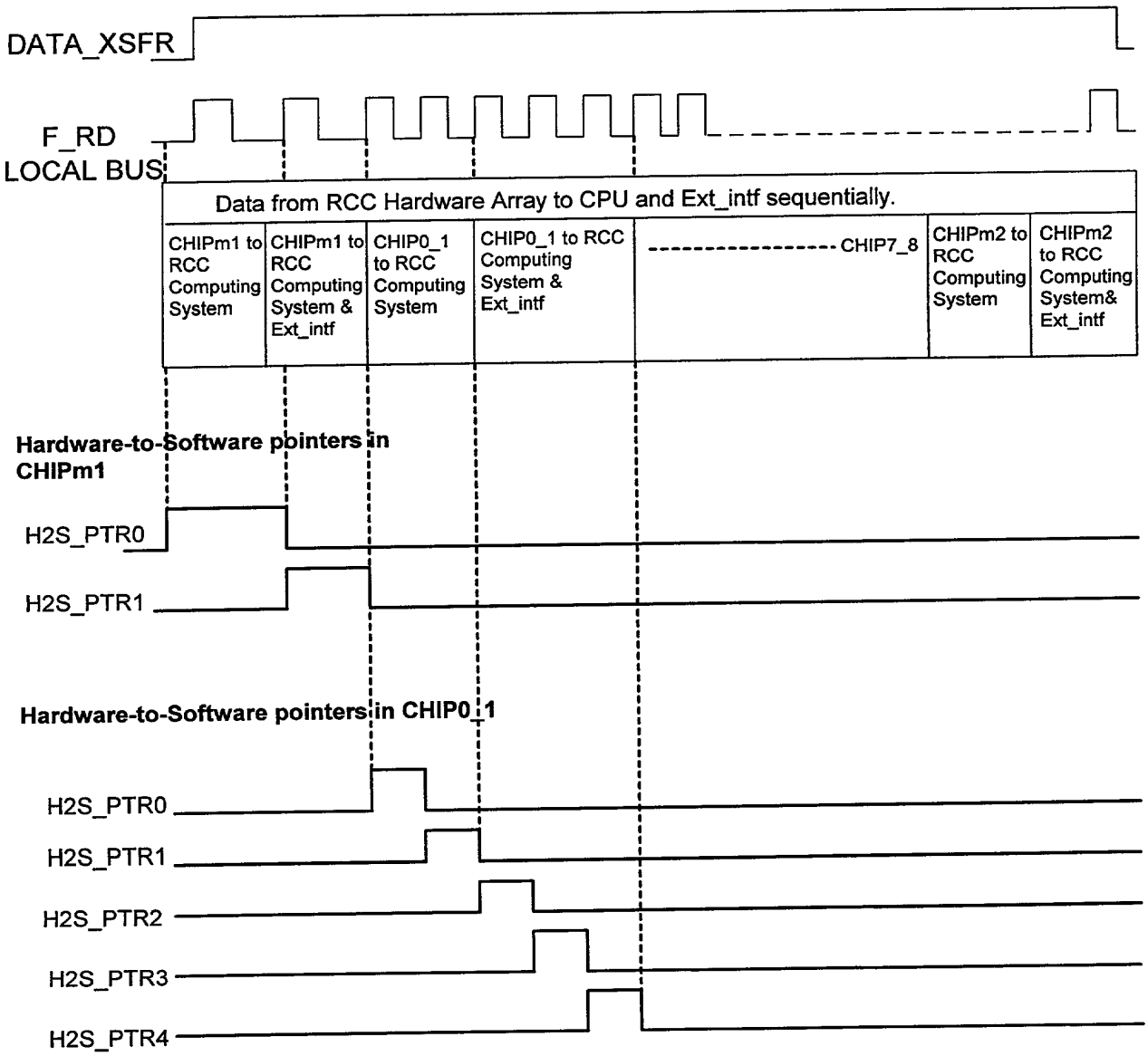


FIG. 73

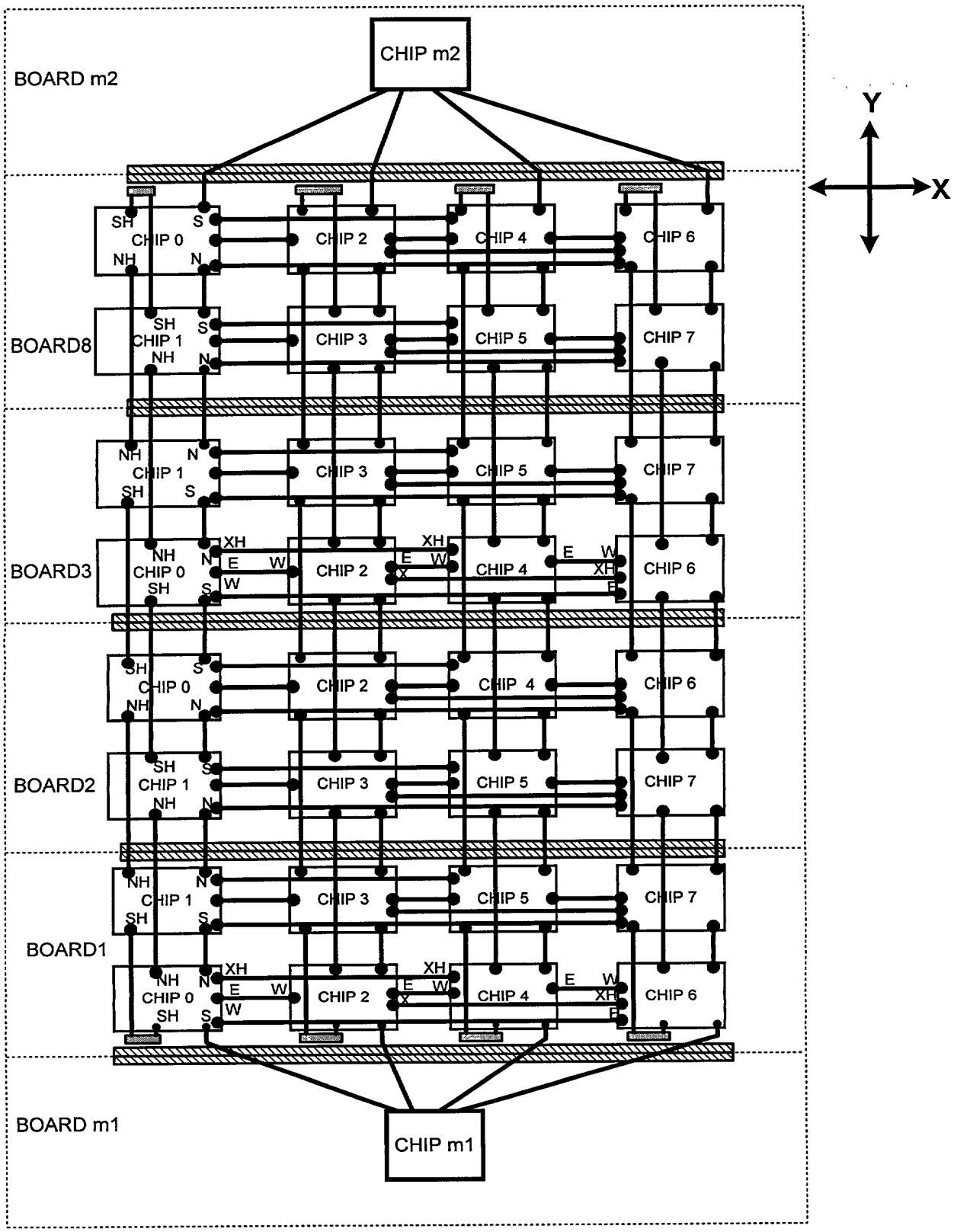

CHIP m2
BOARD m2
Y
X
SH
S
CHIP 0
NH
N
CHIP 2
CHIP 4
CHIP 6
BOARD8
CHIP 1
CHIP 3
CHIP 5
CHIP 7
XH
E
W
BOARD3
BOARD2
BOARD1
BOARD m1
CHIP m1

FIG. 74

SHIFT REGISTER

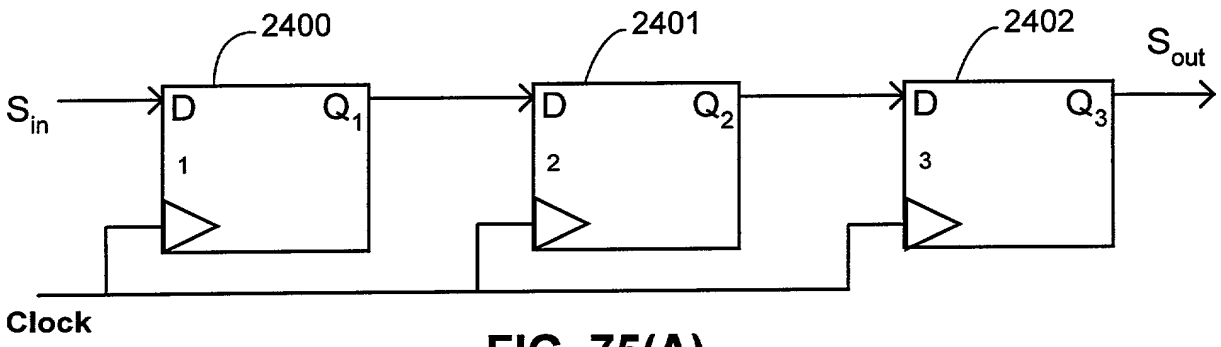


FIG. 75(A)

HOLD TIME ASSUMPTION FOR SHIFT REGISTER

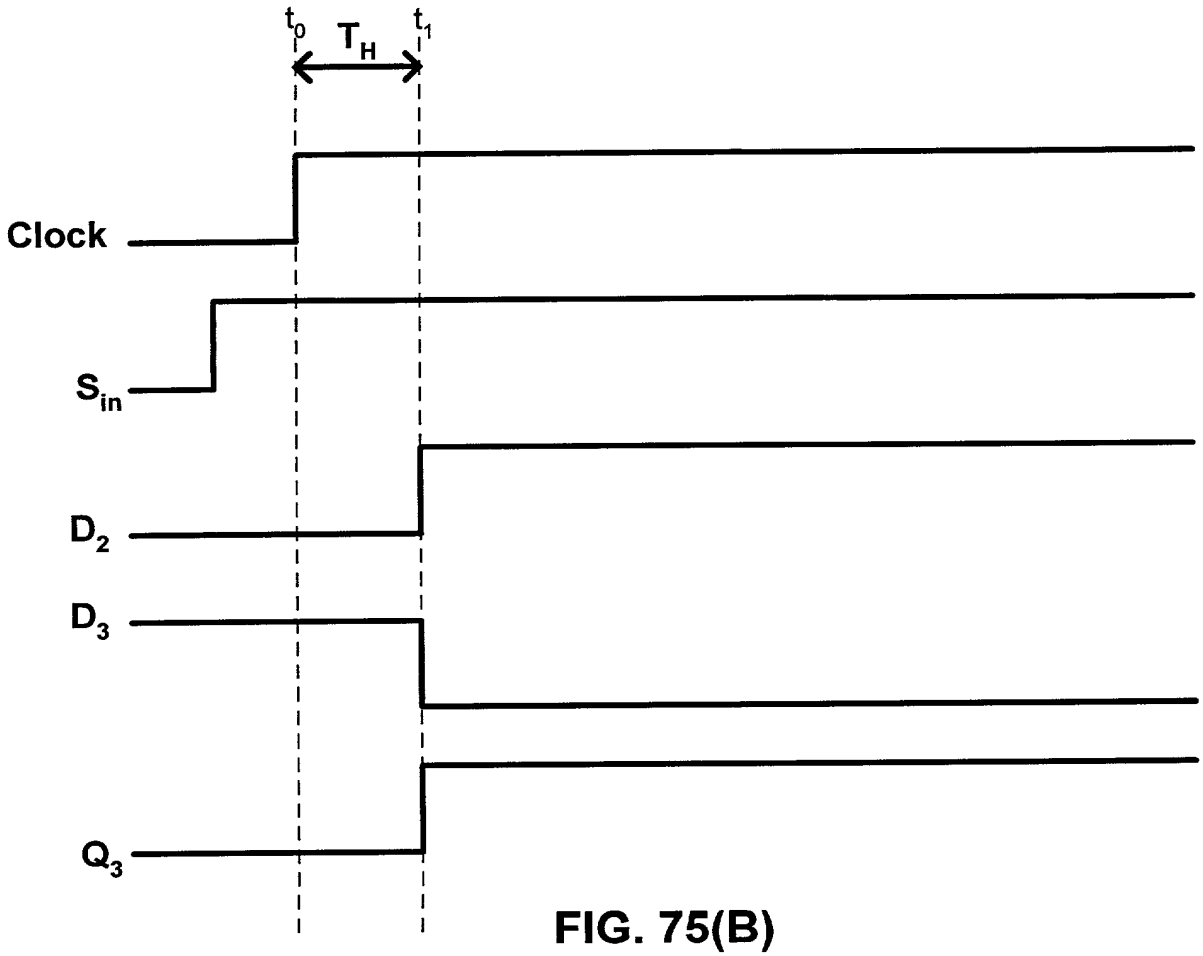


FIG. 75(B)

MULTIPLE FPGA MAPPING FOR SHIFT REGISTER

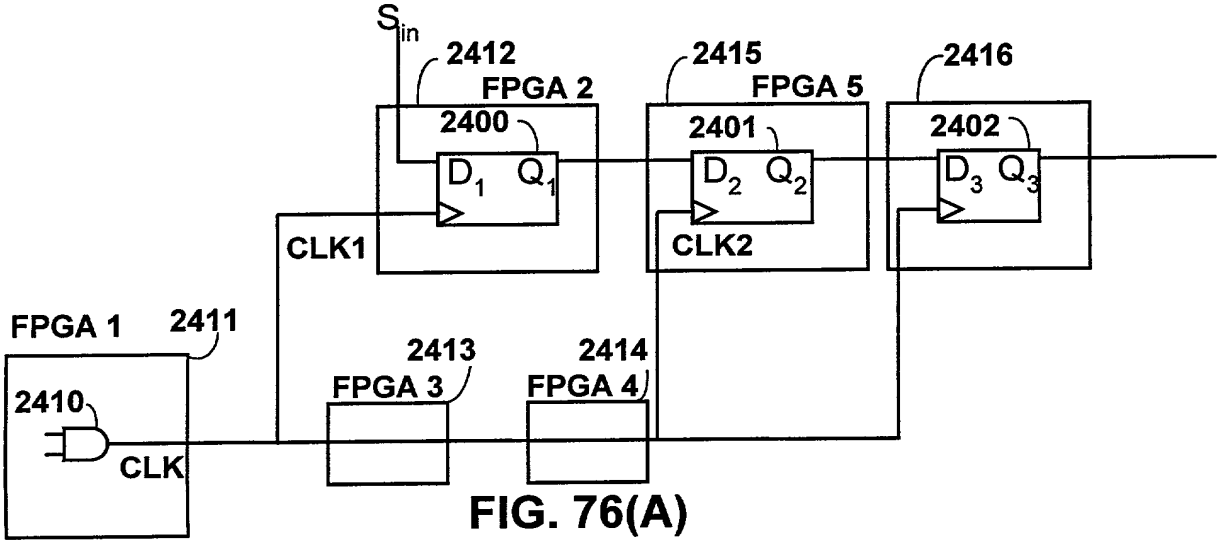


FIG. 76(A)

HOLD TIME VIOLATION BY LONG CLOCK SKEW

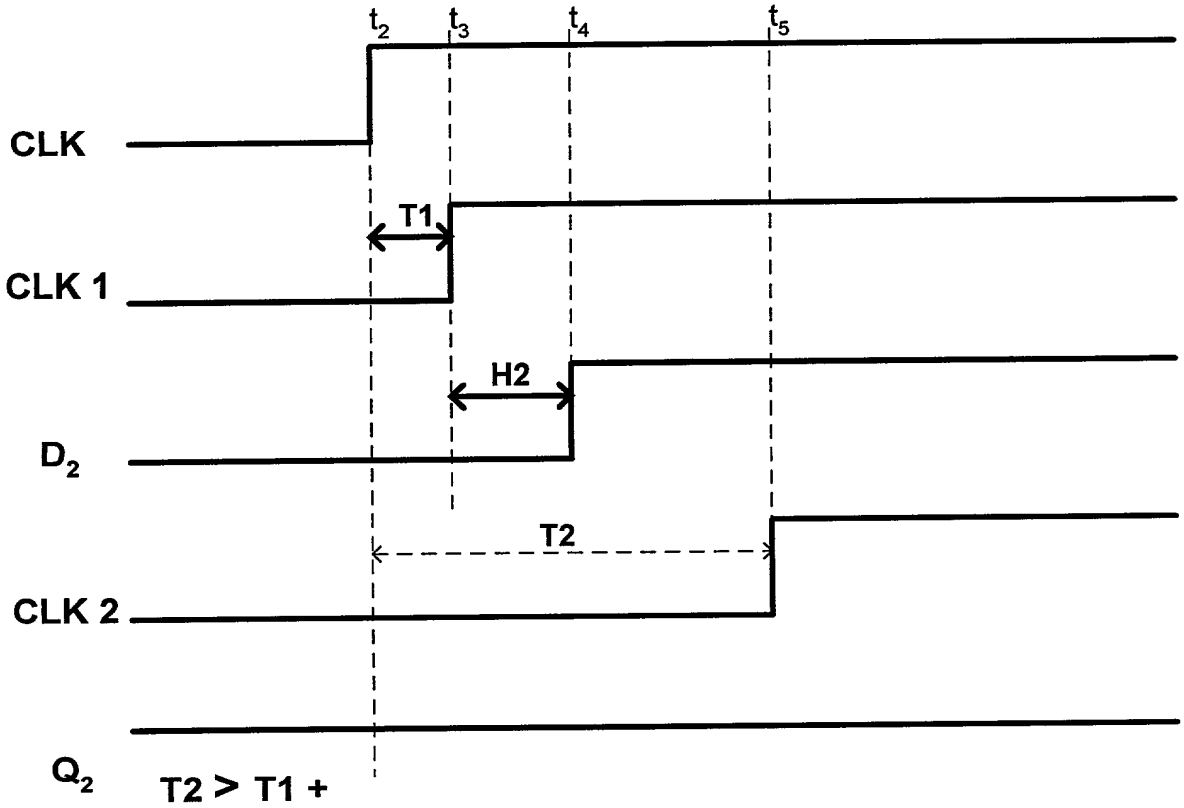


FIG. 76(B)

CLOCK GLITCH PROBLEM

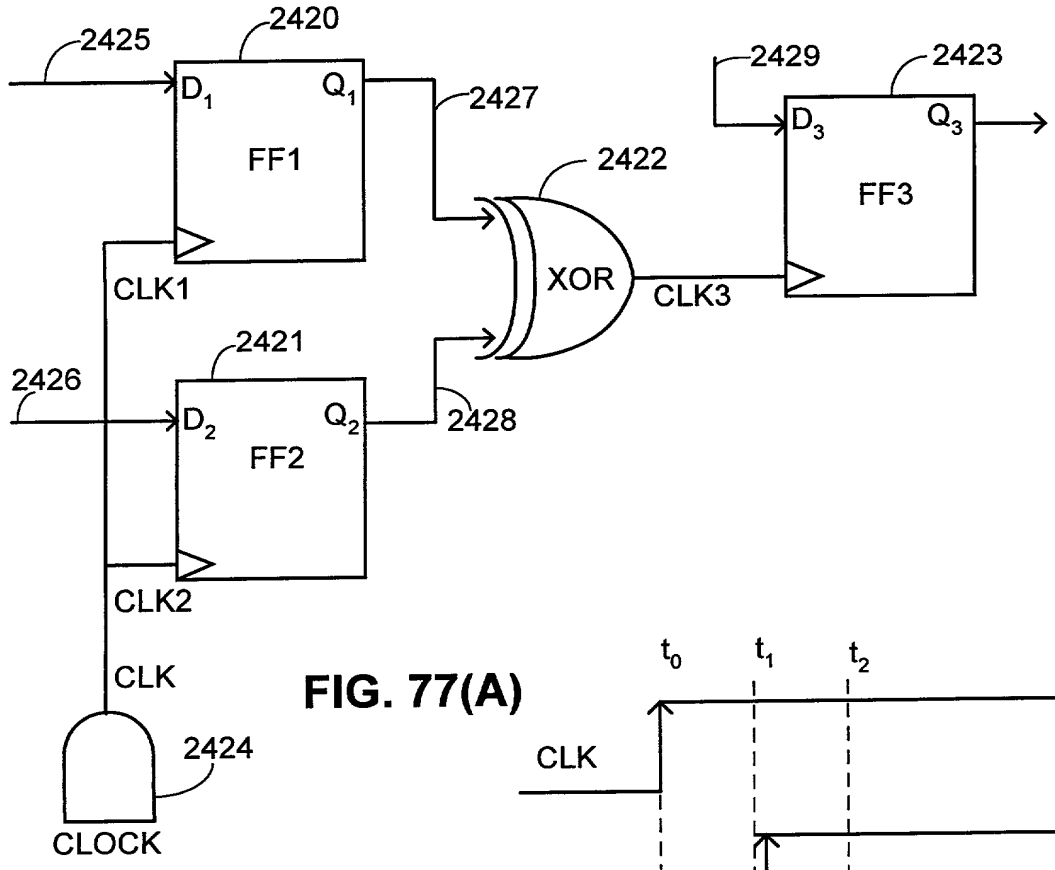


FIG. 77(A)

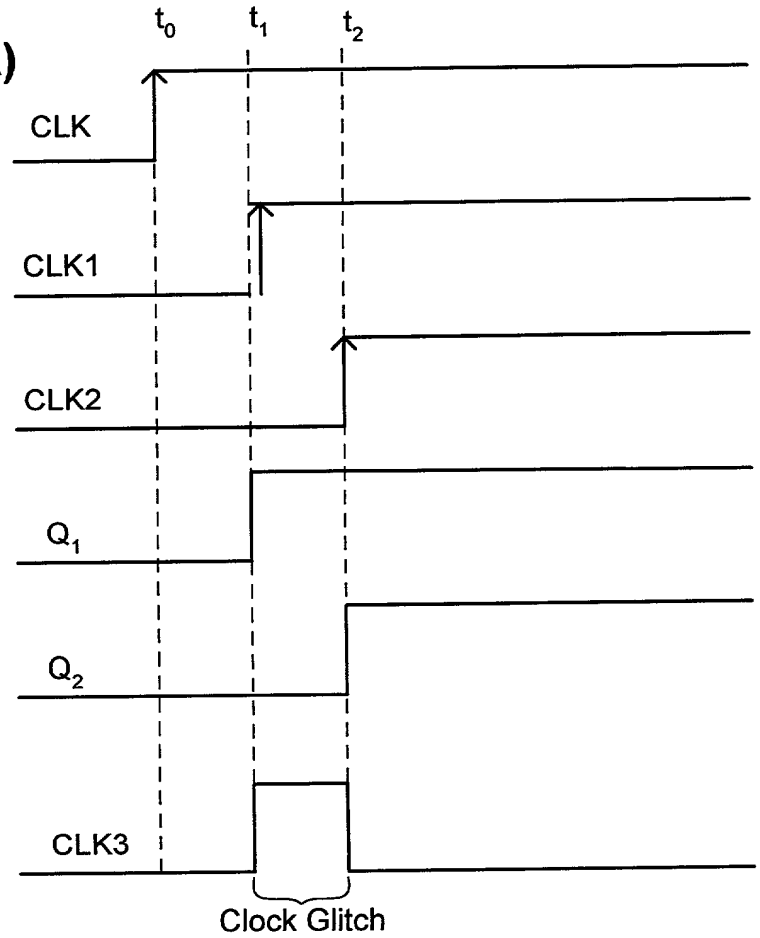
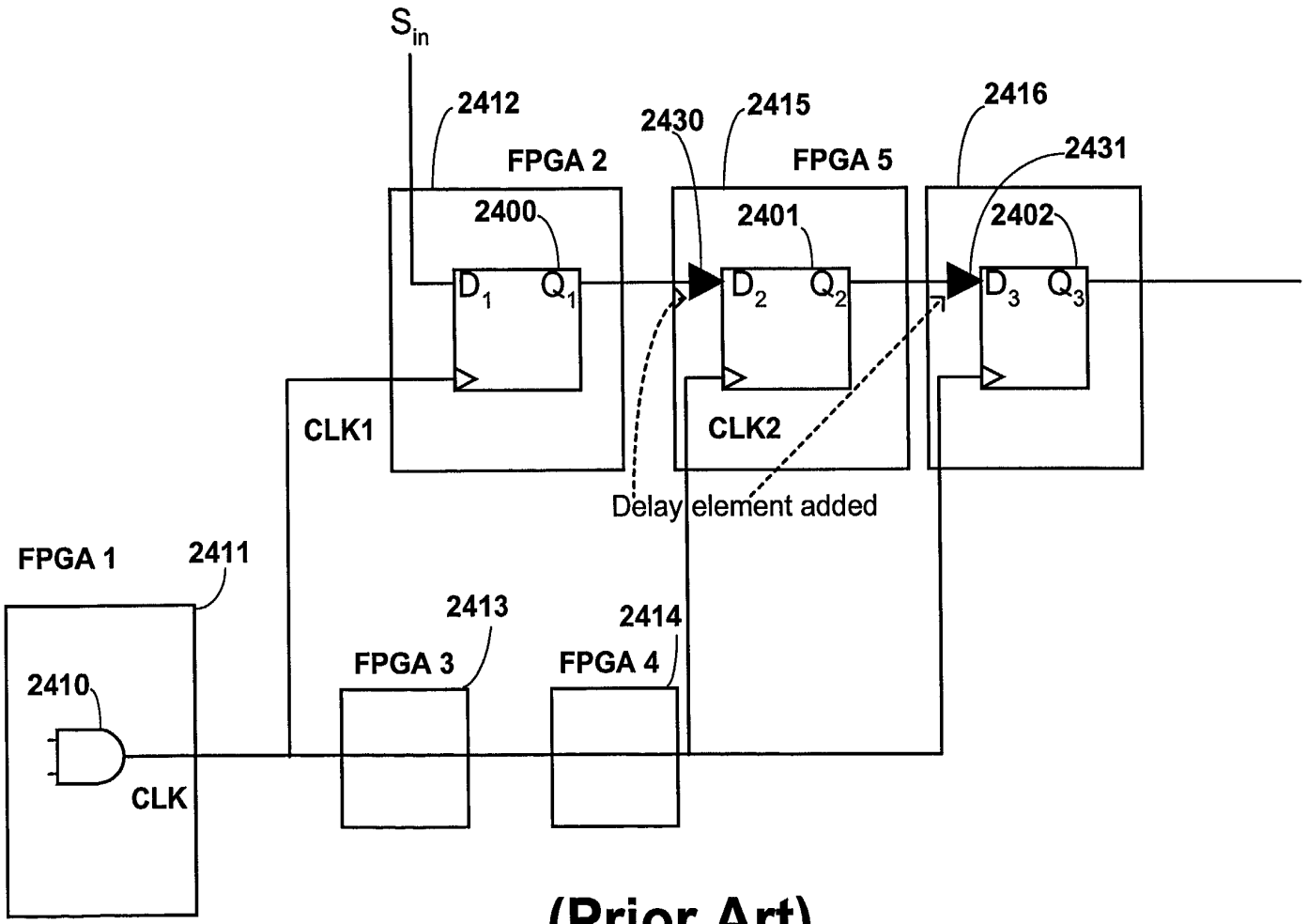


FIG. 77(B)

TIMING ADJUSTMENT BY ADDING DELAY

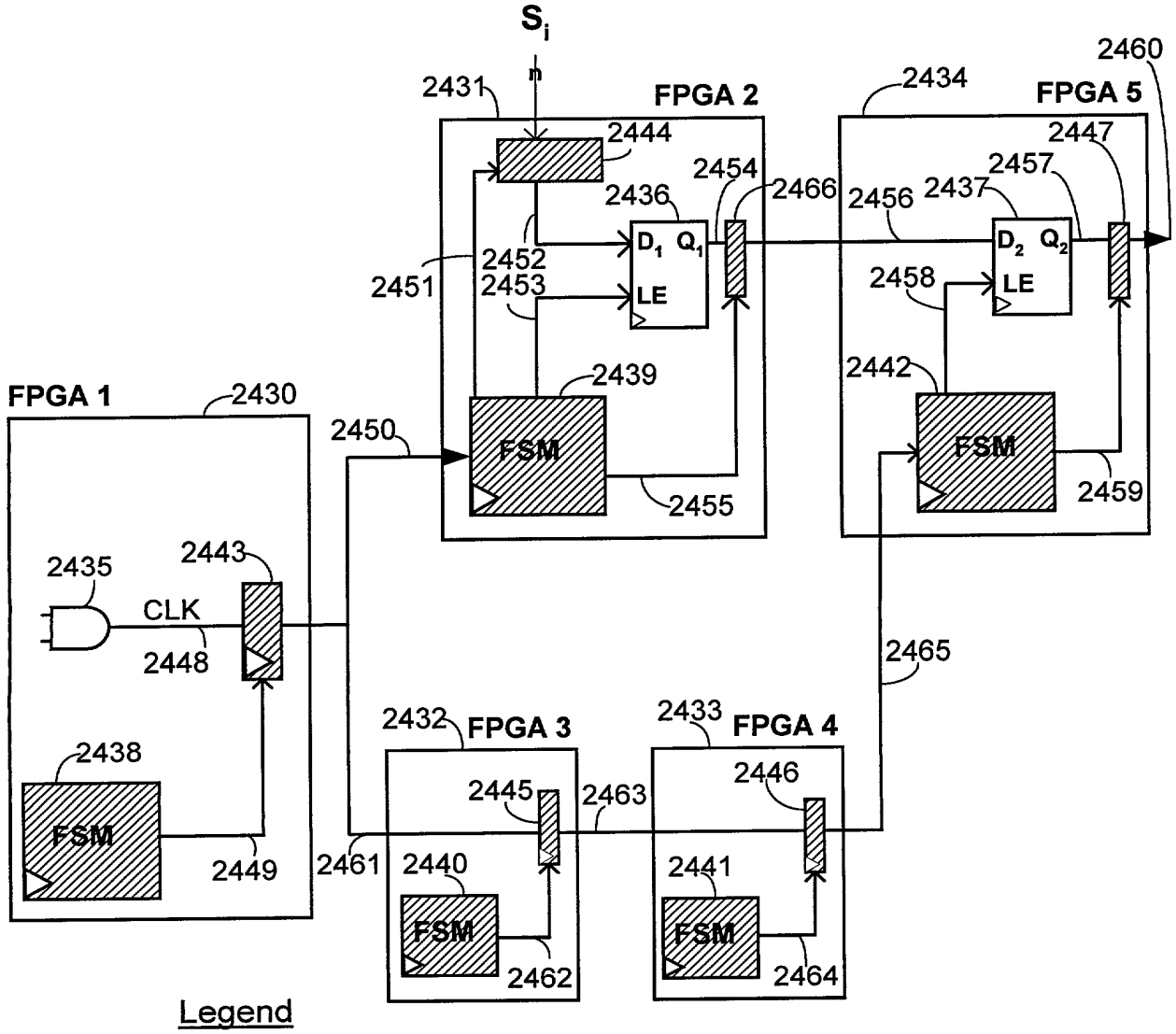


(Prior Art)

FIG. 78

Replacement Sheet

GLOBAL RETIMING



Legend

Controlled by the global reference clock.

FSM and I/O registers for retiming control.

(Prior Art)

FIG. 79

TIGF LATCH

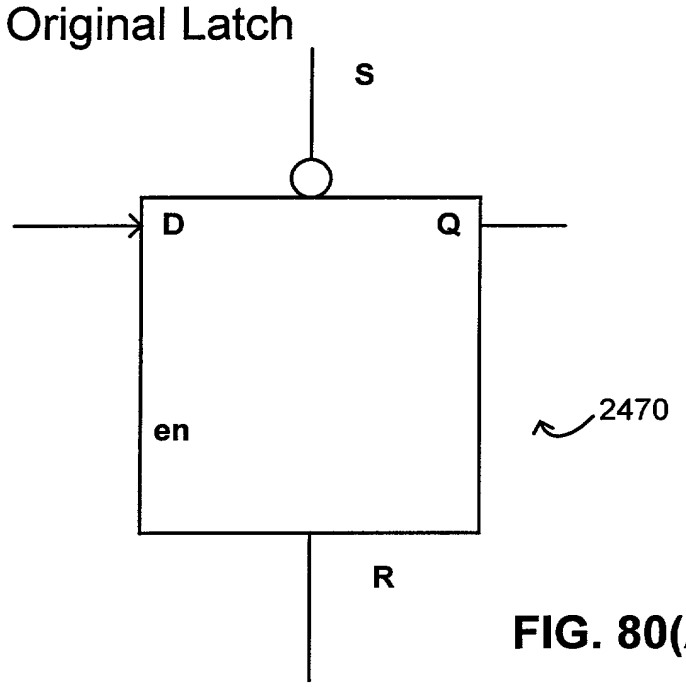


FIG. 80(A)

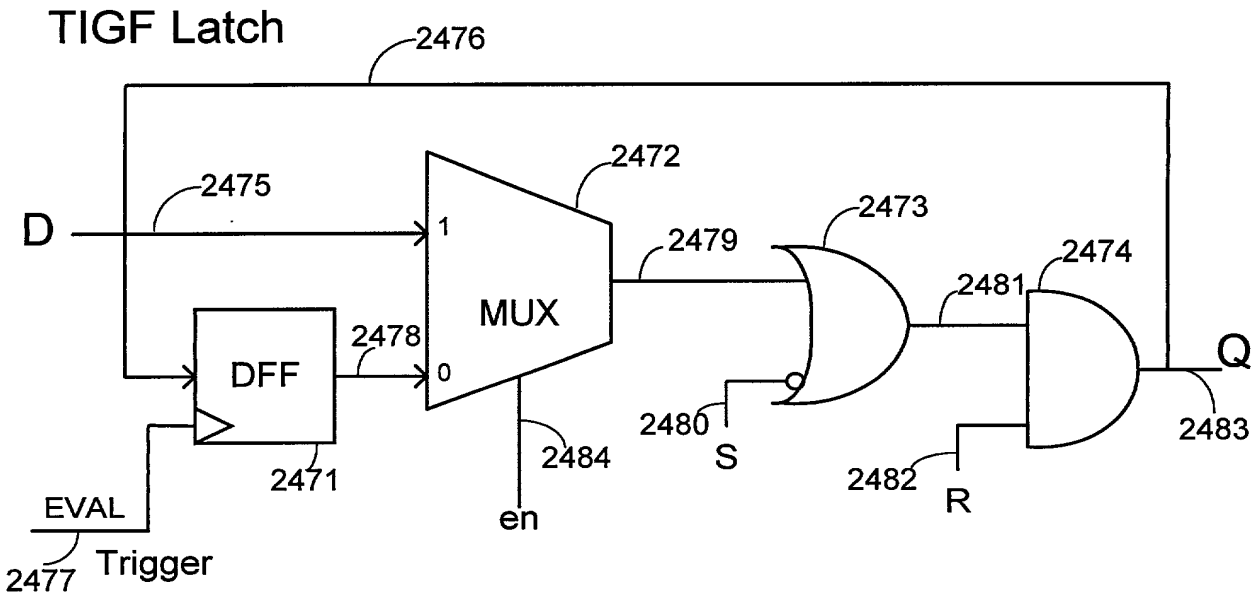


FIG. 80(B)

TIGF DFF

Original DFF

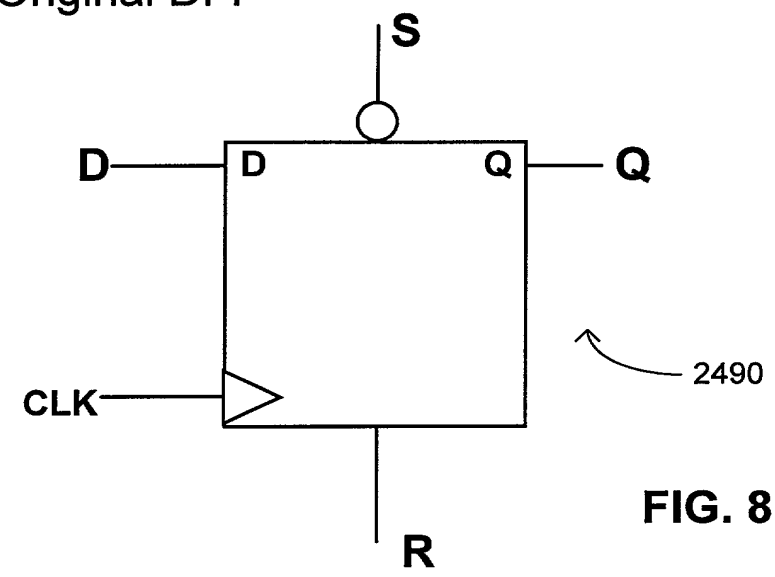


FIG. 81(A)

TIGF DFF and Edge Detector

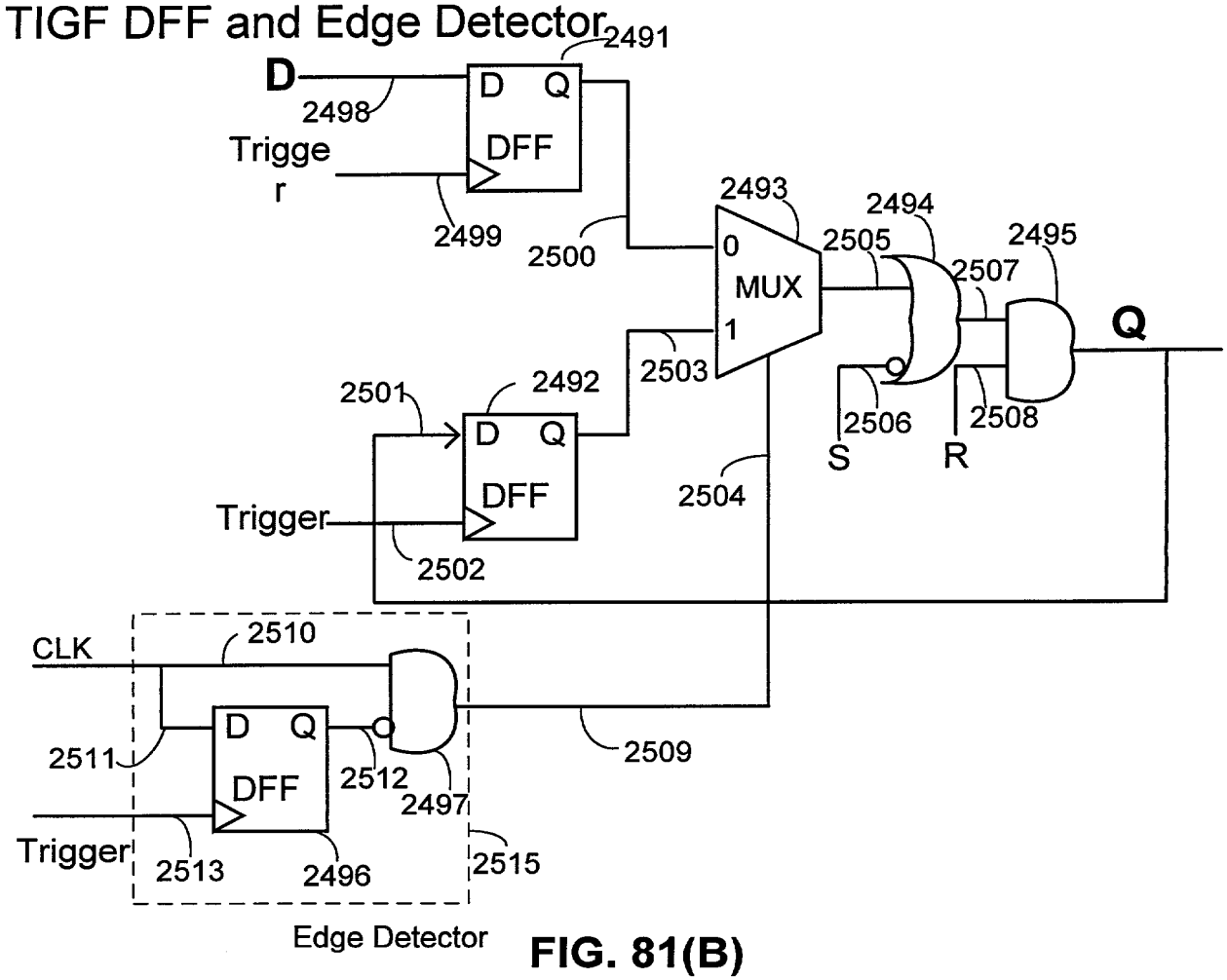


FIG. 81(B)

TOOTOT "k2T00660

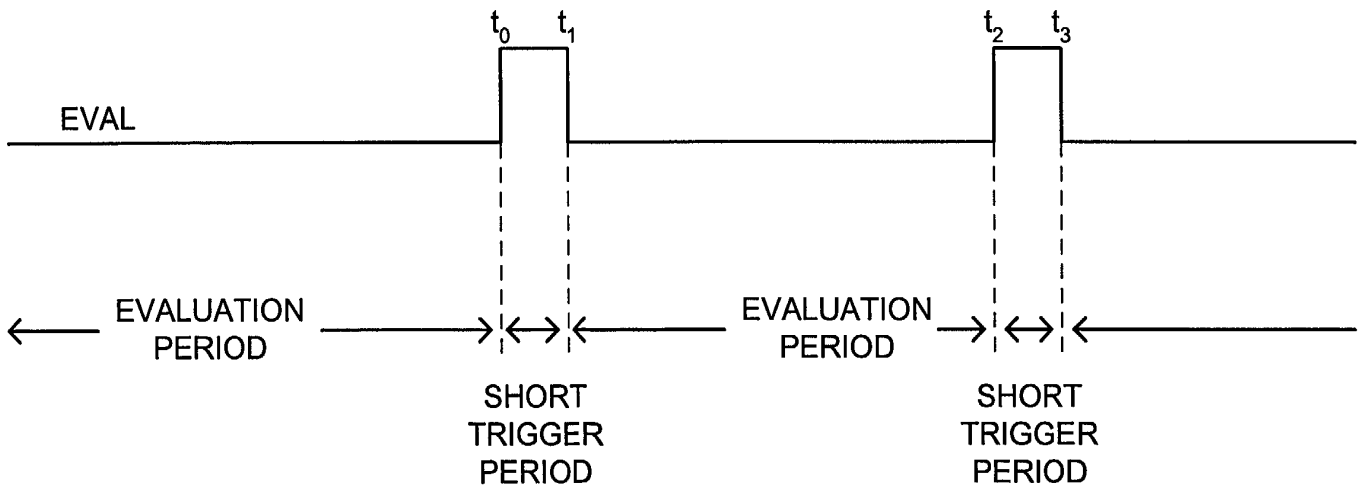


FIG. 82

RCC System

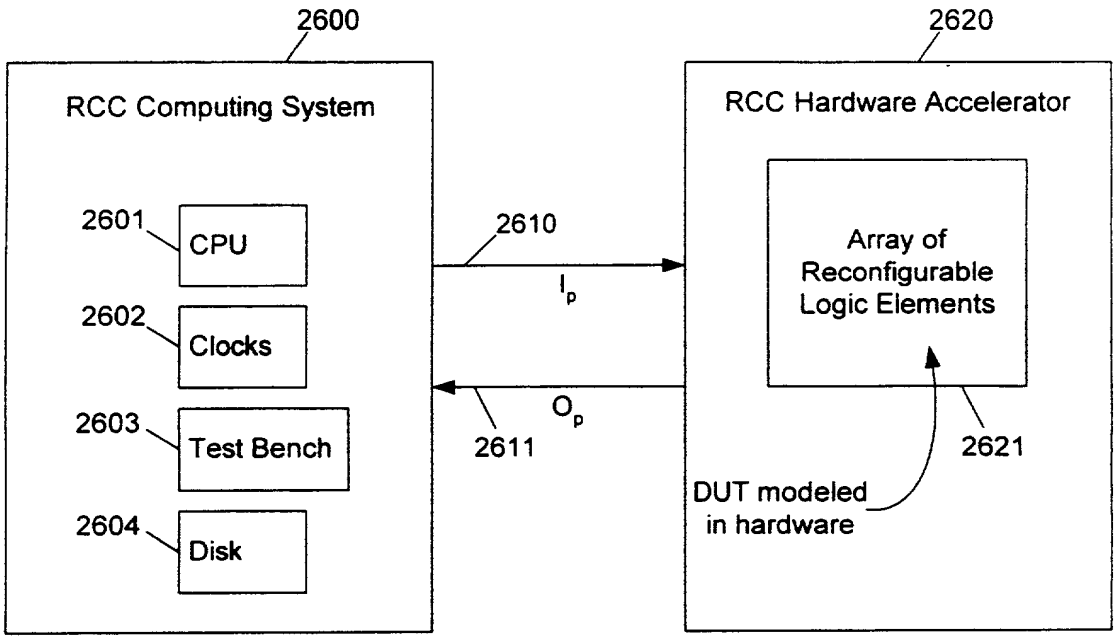


FIG. 83

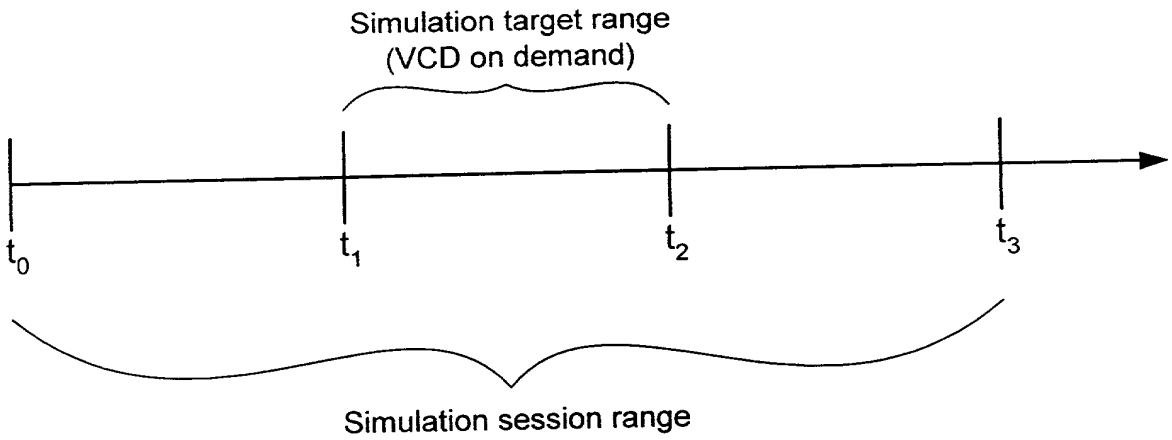

Simulation target range
(VCD on demand)
t_0
t_1
t_2
t_3
Simulation session range

FIG. 84

SINGLE-ROW FPGA PER BOARD

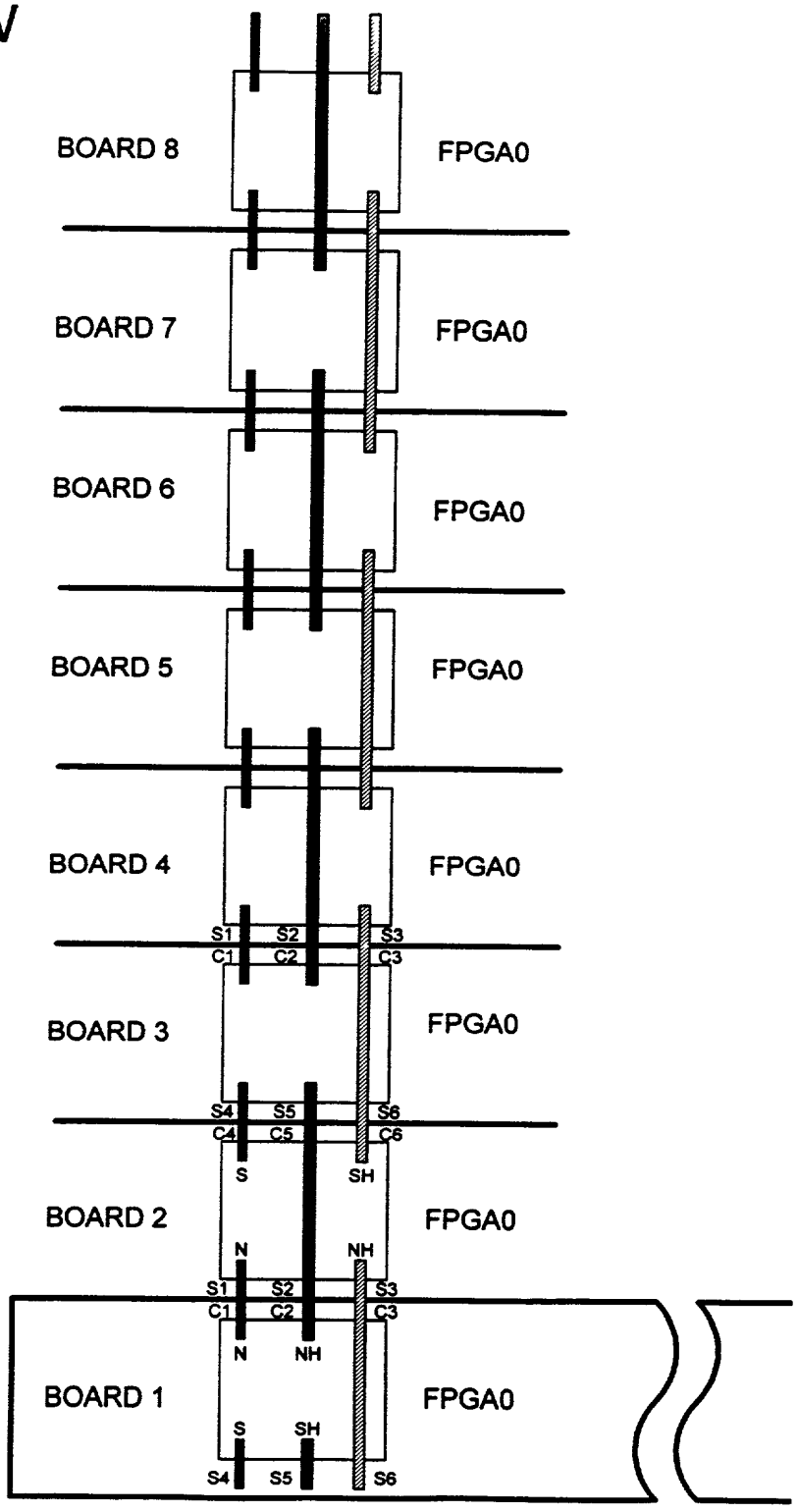


FIG. 85

TWO-ROW FPGA PER BOARD

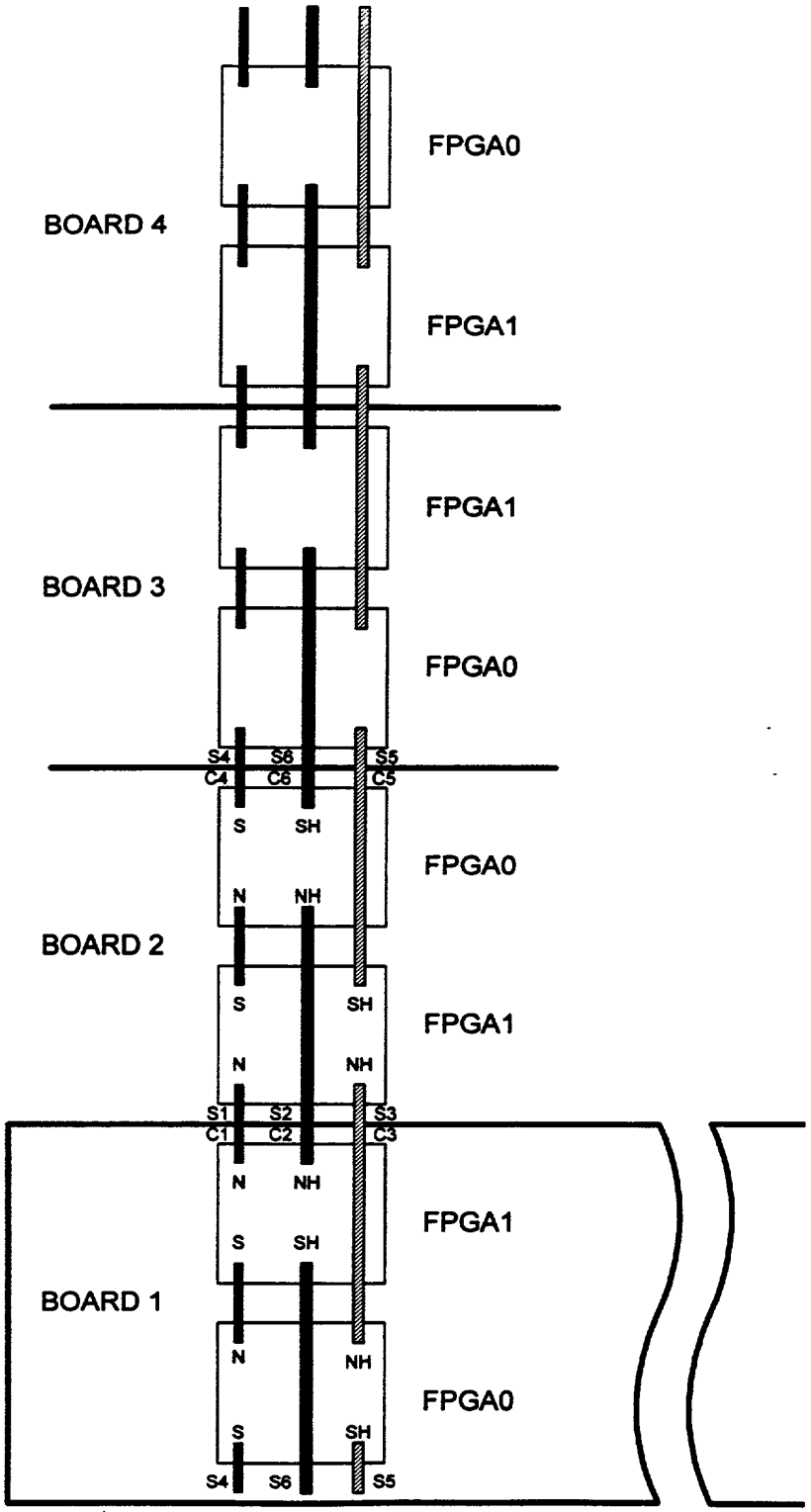


FIG. 86

THREE-ROW FPGA PER BOARD

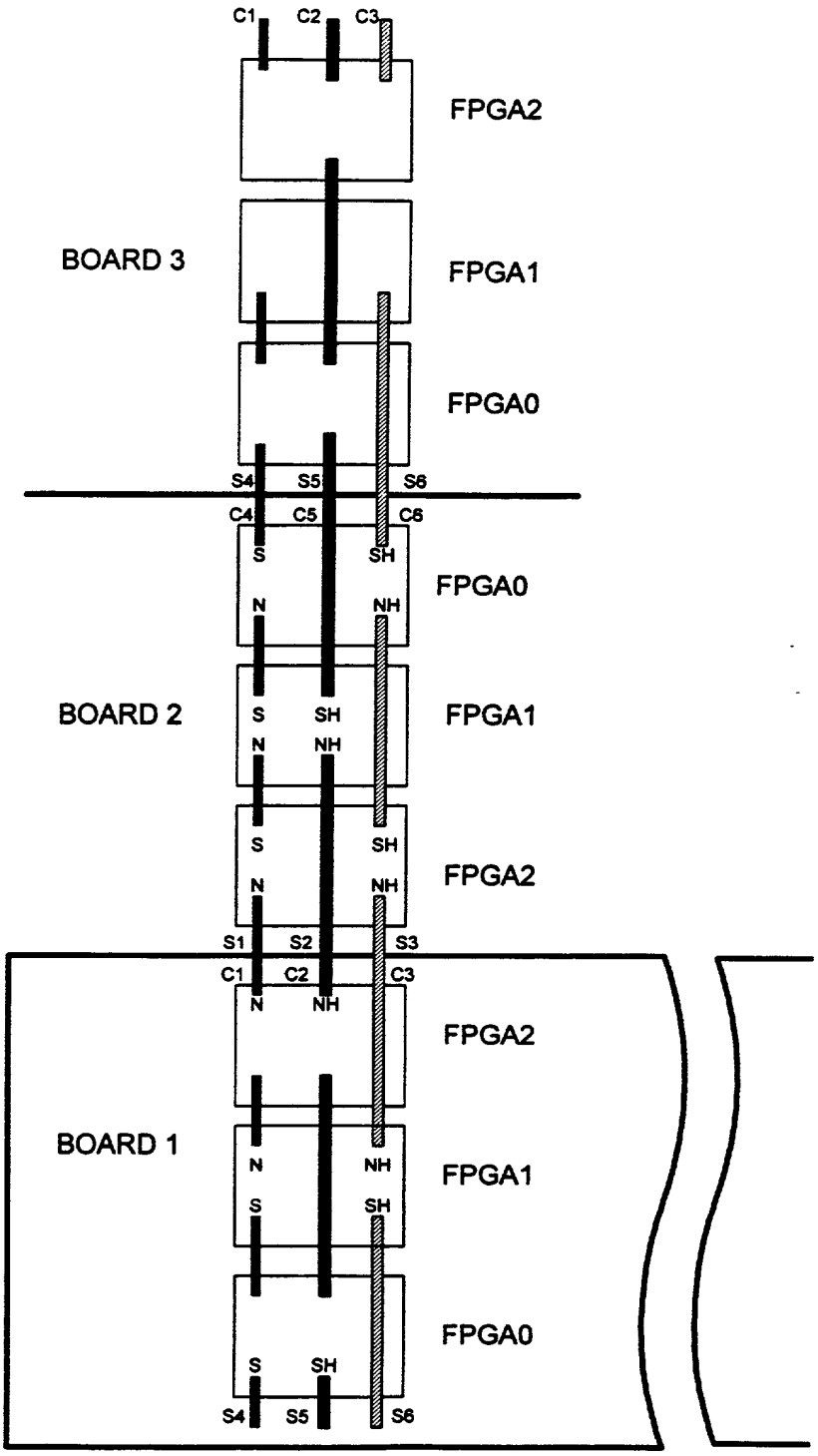


FIG. 87

FOUR-ROW FPGA PER BOARD

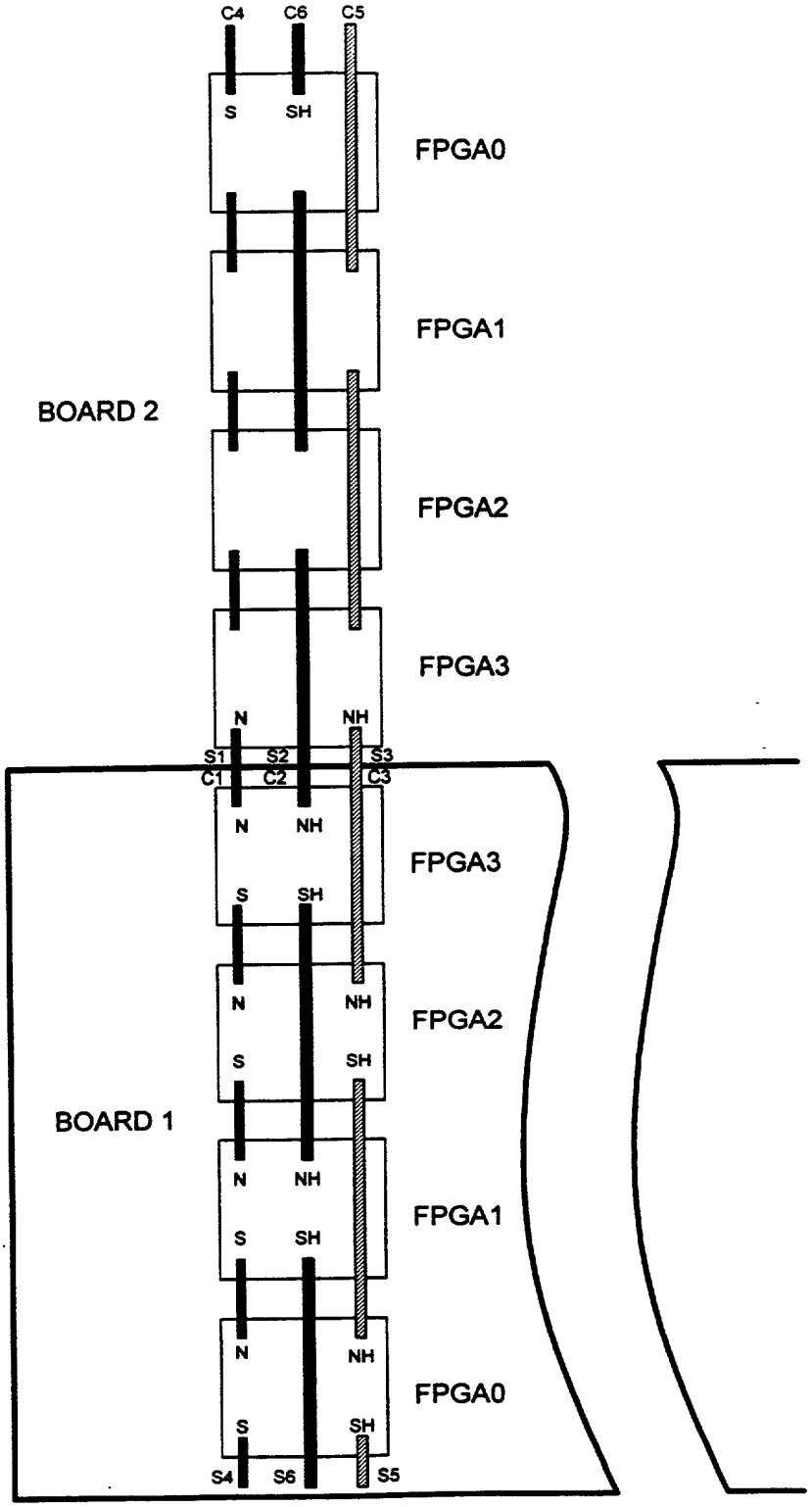


FIG. 88

INTERCONNECT FOR THREE-ROW PER BOARD

I/O Signals	Odd Board	Even Board	Common Board
	Connector-Group Pin-position	Connector-Group Pin-position	Connector-Group Pin-position
FPGA2_N	C1	S1	C1, S1
FPGA2_NH	C2	S3	C2, S3
FPGA1_NH	C3	S2	C3, S2
FPGA0_S	S4	C4	C4, S4
FPGA0_SH	S5	C6	C6, S5
FPGA1_SH	S6	C5	C5, S6

FIG. 89

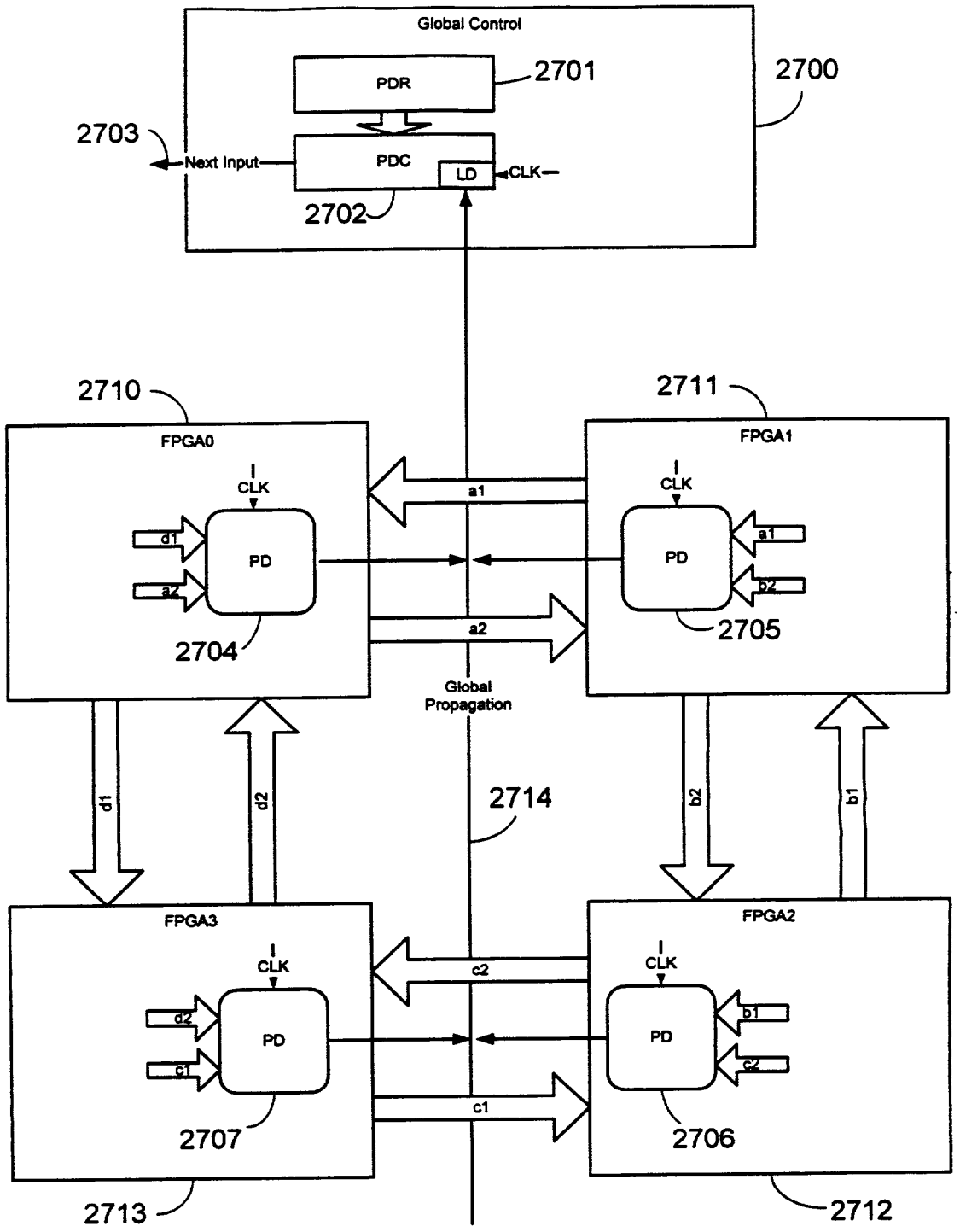

Global Control
PDR
2701
2700
2703
Next Input
PDC
LD
CLK
2702
2710
2711
FPGA0
FPGA1
CLK
d1
a2
PD
a1
b2
2704
2705
a1
a2
Global
Propagation
2714
d1
d2
b2
b1
FPGA3
FPGA2
c2
d2
c1
b1
2707
2706
c1
2713
2712

FIG. 90

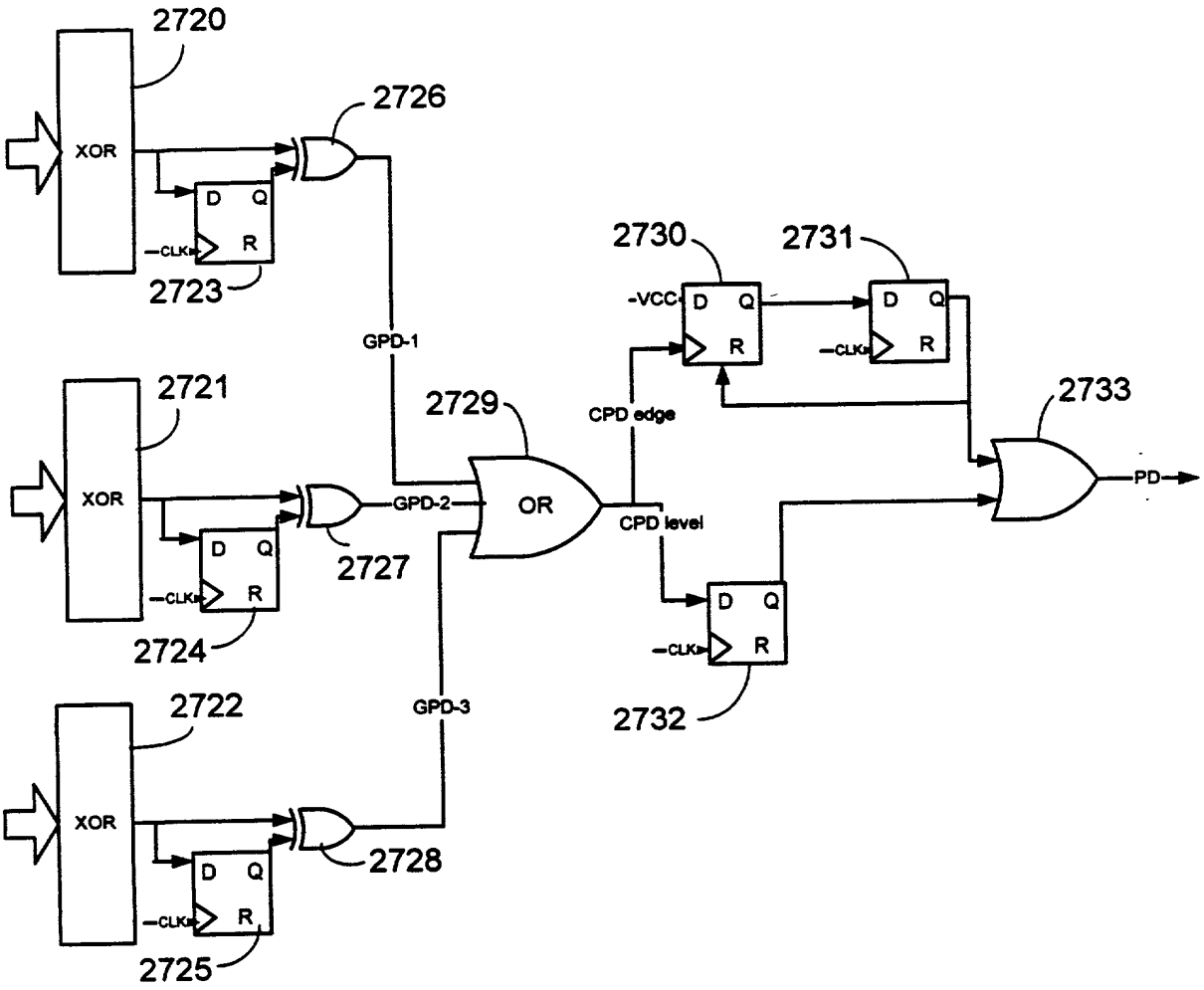

2720
XOR
2726
D Q
CLK R
2723
GPD-1
2721
XOR
2729
GPD-2
OR
2727
D Q
CLK R
2724
2722
XOR
GPD-3
2728
D Q
CLK R
2725
2730
-VCC
D Q
R
2731
D Q
CLK R
CPD edge
2733
PD
CPD level
D Q
CLK R
2732

FIG. 91

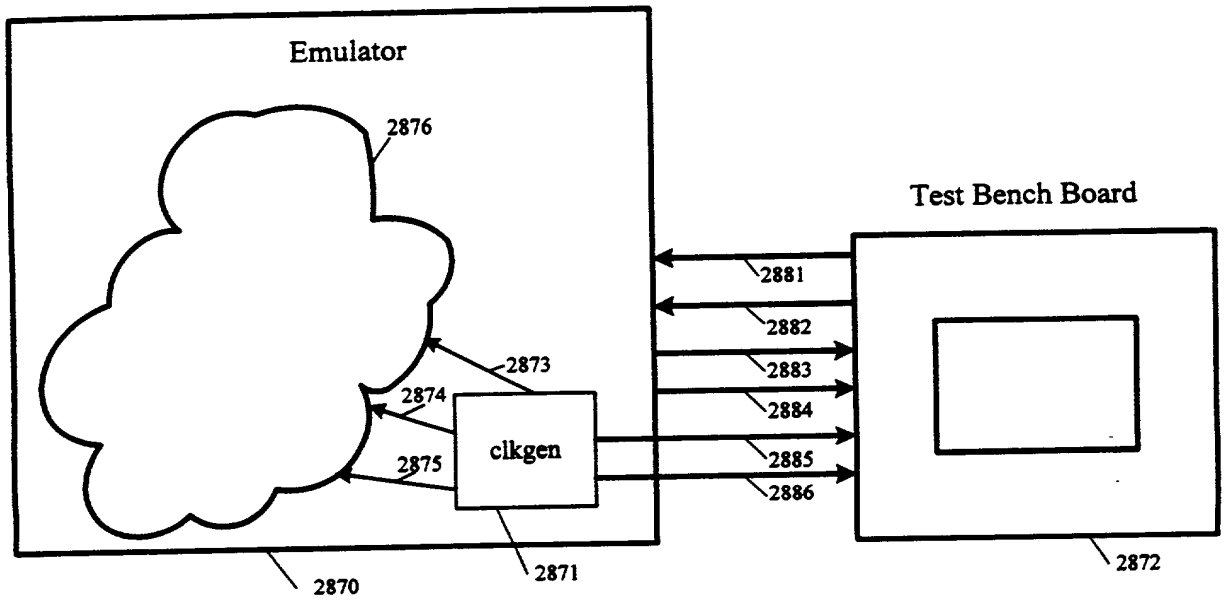

Emulator
2876
Test Bench Board
2881
2882
2883
2884
2885
2886
2873
2874
clkgen
2875
2870
2871
2872

FIG. 92

Clock Specification

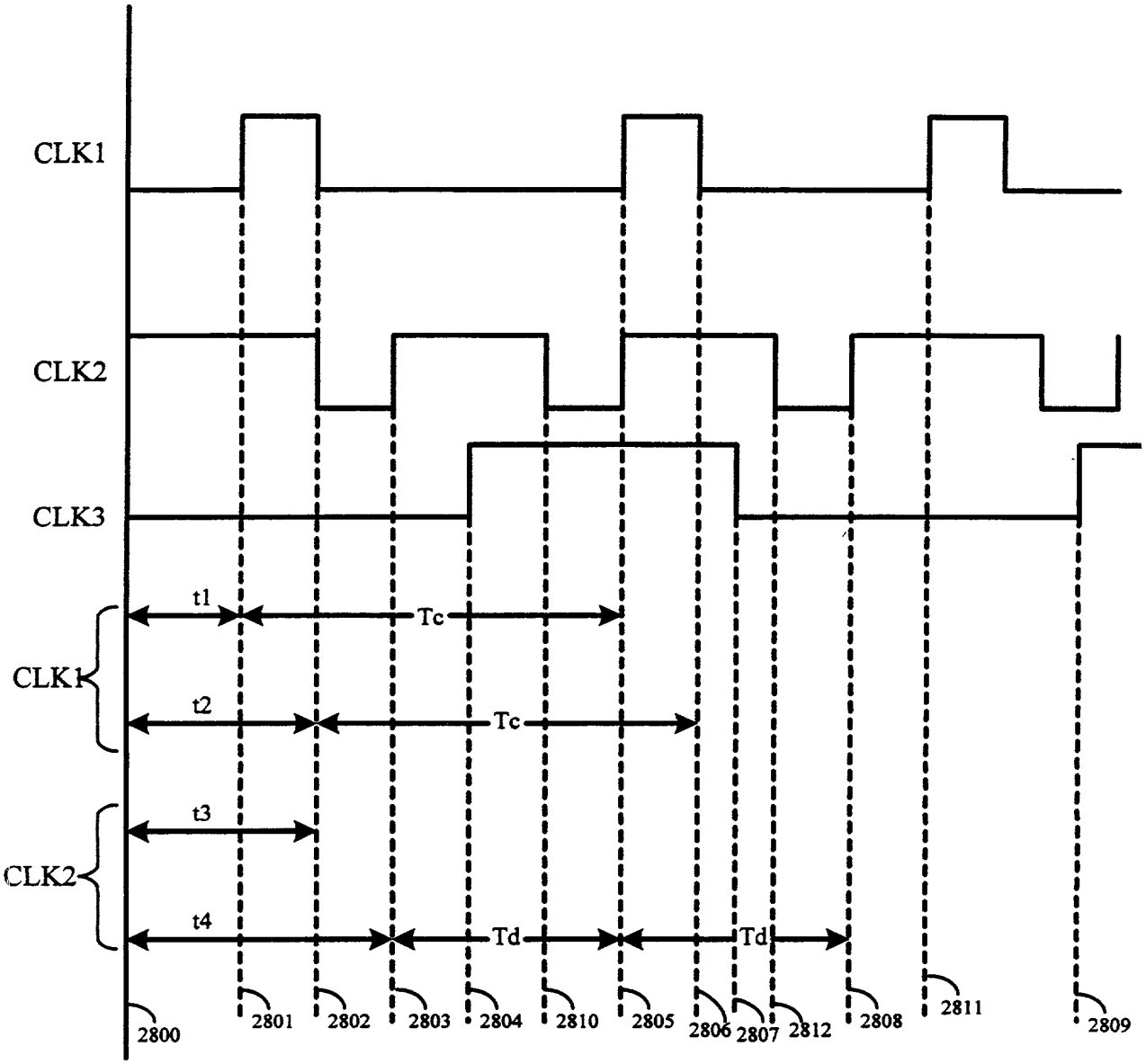


FIG. 93

Clock Generation Scheduler w/ Slices

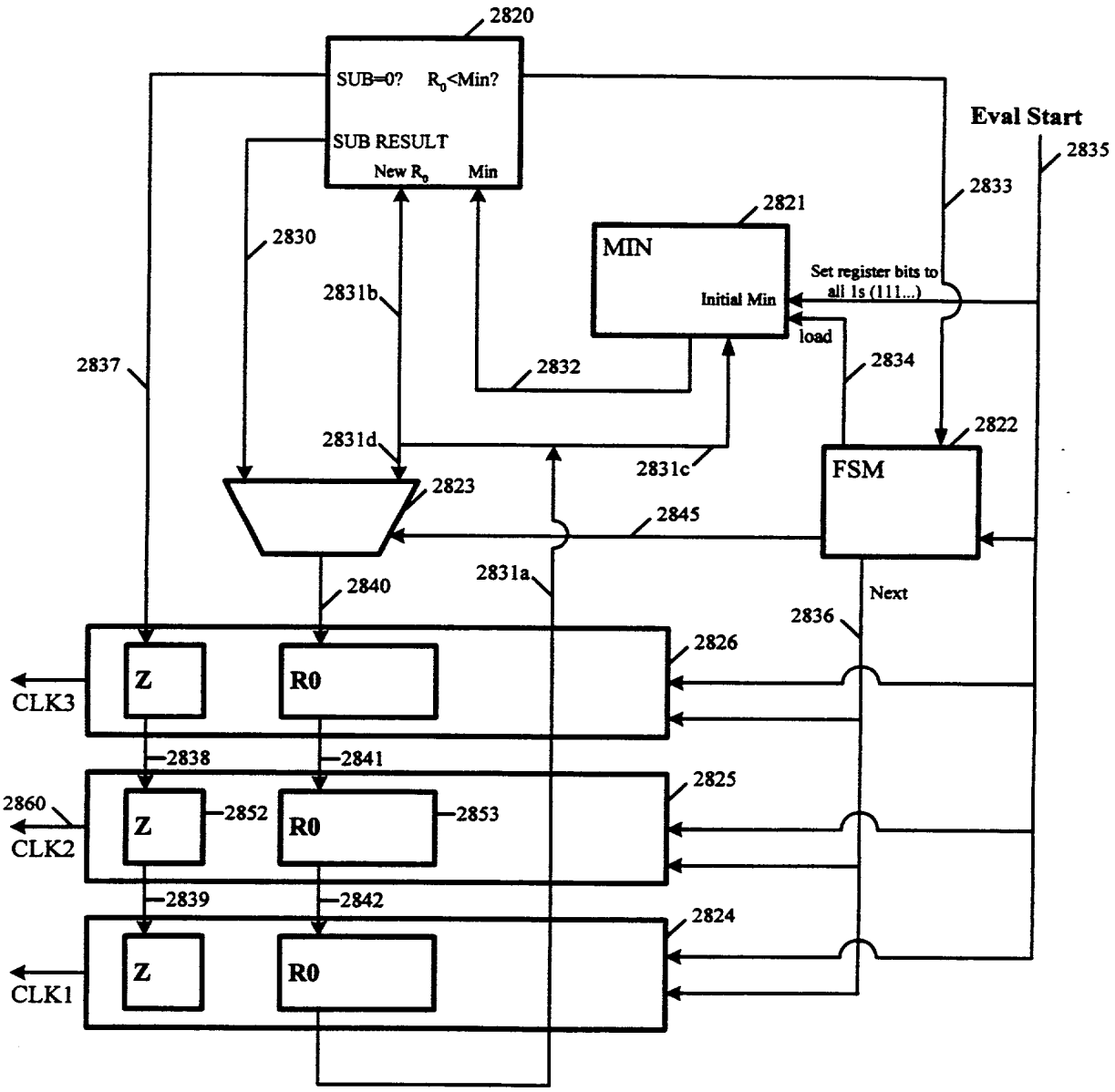


FIG. 94

Clock Generation Slice

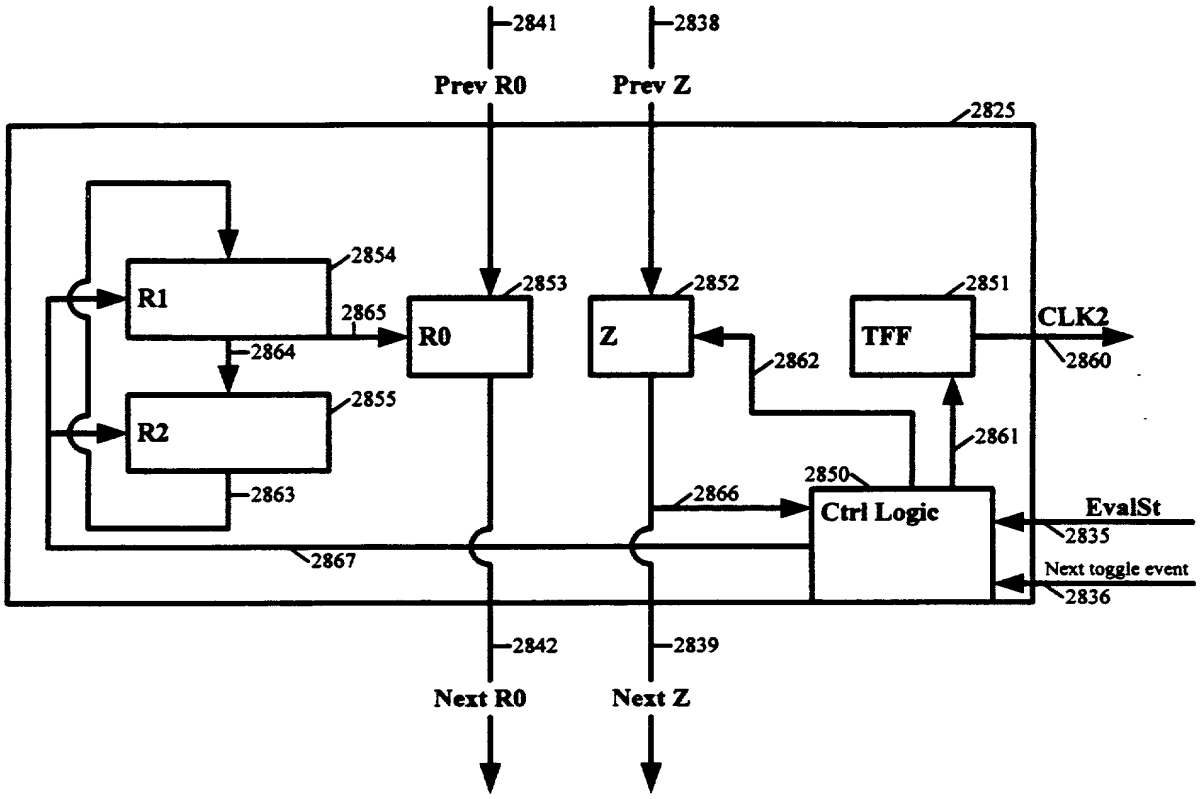


FIG. 95

Clock Generation Scheduler and Slices

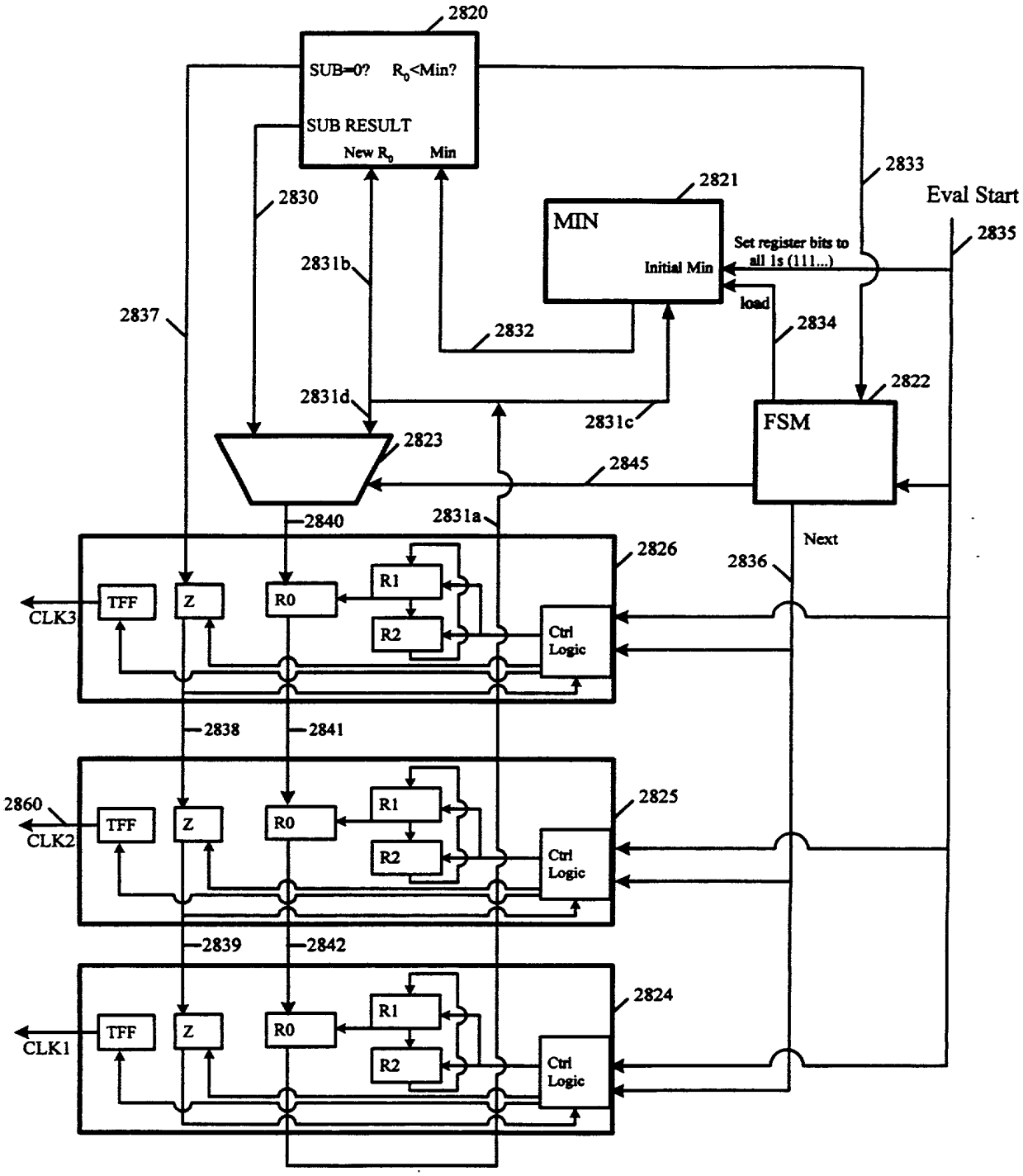


FIG. 96

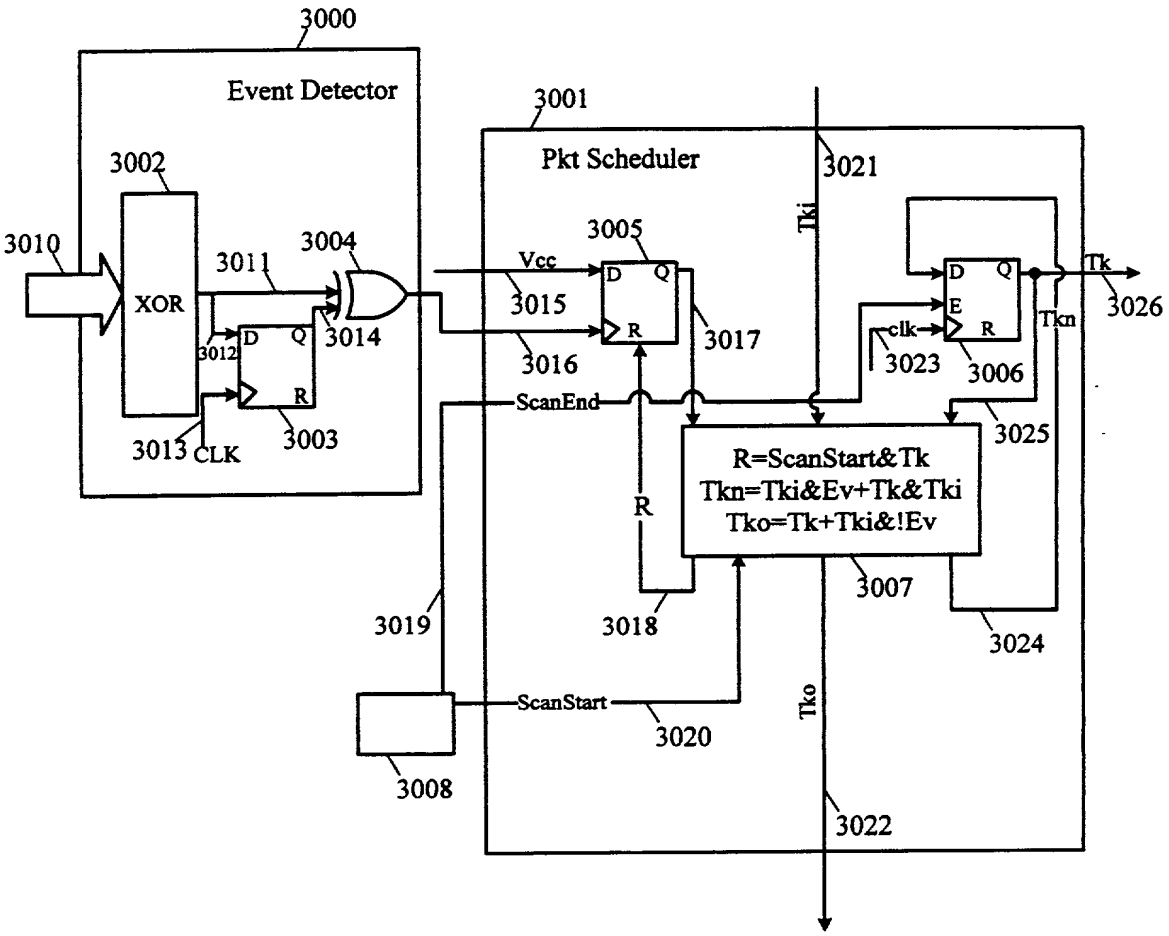

3000
Event Detector
3001
Pkt Scheduler
3002
3010
XOR
3011
3004
3012
D
Q
3014
R
3003
3013
CLK
3005
Vcc
3015
3016
3017
3021
Tki
3023
clk
E
3006
Tkn
Tk
3026
ScanEnd
3025
R=ScanStart&Tk
Tkn=Tki&Ev+Tk&Tki
Tko=Tk+Tki&!Ev
R
3007
3018
3019
3024
ScanStart
3020
Tko
3008
3022

FIG. 97

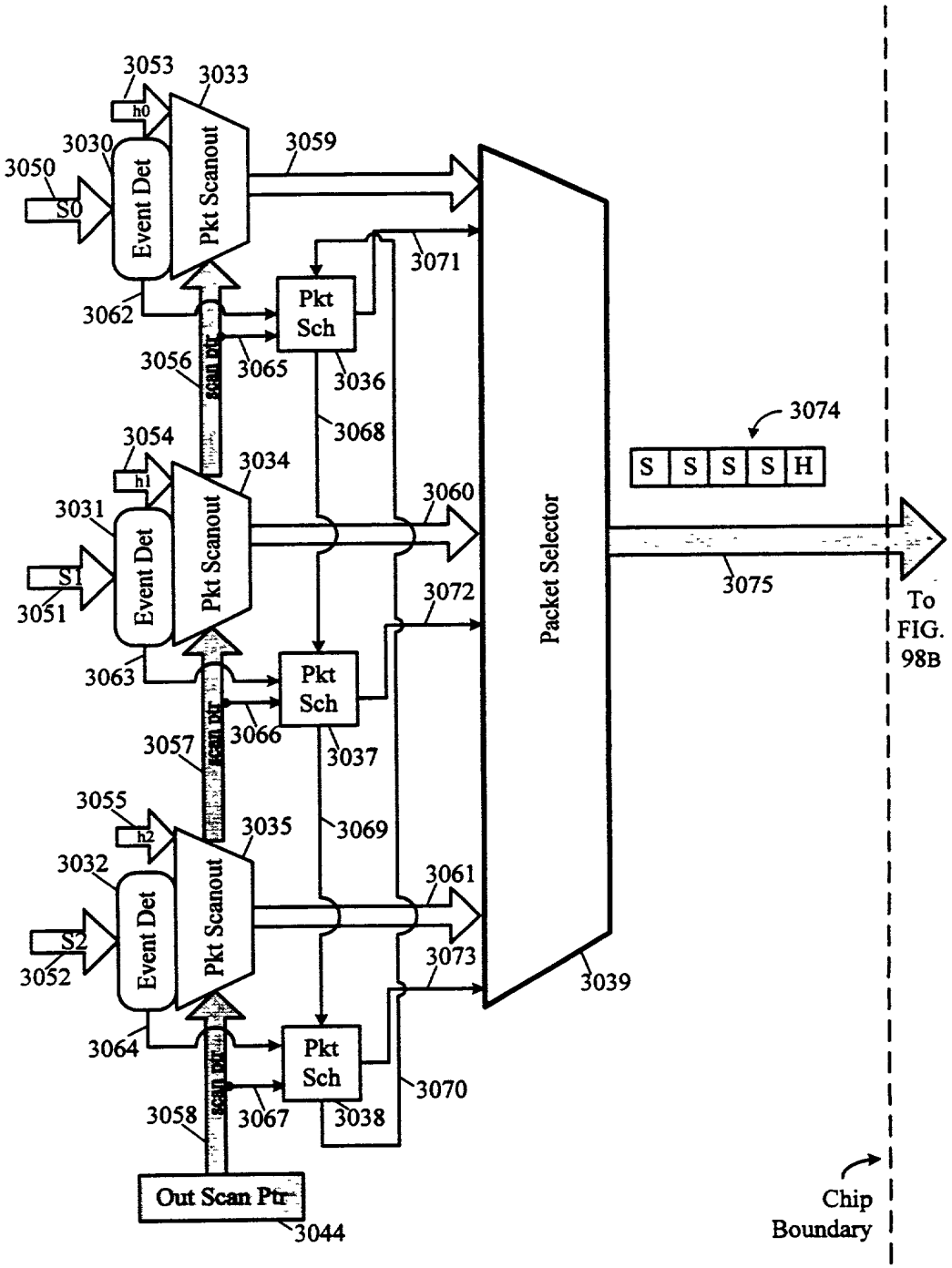

3053
3033
h0
3030
3050
S0
Event Det
Pkt Scanout
3059
3071
3062
Pkt Sch
3056
Scan Ptr
3065
3036
3068
3054
3034
h1
3031
3060
S1
3051
Event Det
Pkt Scanout
3072
3063
Pkt Sch
3066
3057
Scan Ptr
3037
3055
h2
3035
3069
3032
3061
S2
3052
Event Det
Pkt Scanout
3073
3064
Pkt Sch
3070
3058
Scan Ptr
3067
3038
Out Scan Ptr
3044
Packet Selector
3039
3074
S S S S H
3075
To FIG. 98B
Chip Boundary

FIG. 98A

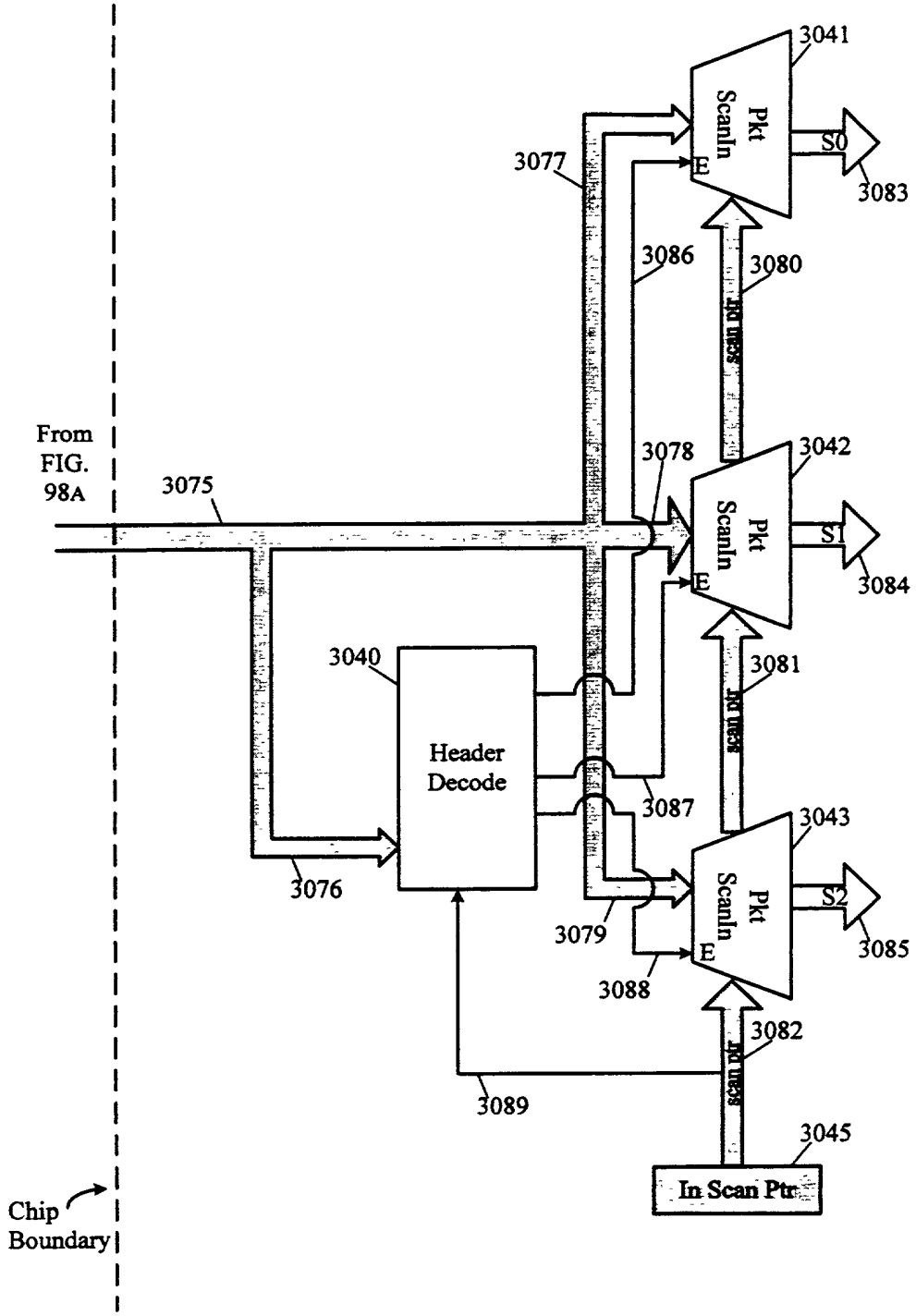

From FIG. 98A
3075
3077
3078
3086
3080
3041
Pkt ScanIn
E
S0
3083
3042
Pkt ScanIn
E
S1
3084
3081
3040
Header Decode
3087
3076
3043
Pkt ScanIn
E
S2
3085
3079
3088
3082
3089
3045
In Scan Ptr
Chip Boundary

FIG. 98B